TANDY LAPTOP COMPUTING VOLUME 5, NUMBER 7 JULY 1988

TERRY KEPNER'S

portable 100

$3.95/CAN $4.95

LAPTOPS SHOOT FOR THE STARS

ROCKET FLIGHT TRACKING

EASY MS-DOS/MODEL 100 FILE TRANSFERS

INPUTING WITH A FLASHLIGHT

DEVICE TIMING WITH YOUR TANDY

TRAVELING SOFTWARE BOOSTER PAK

74724 07 0 186036 8

Announcing:

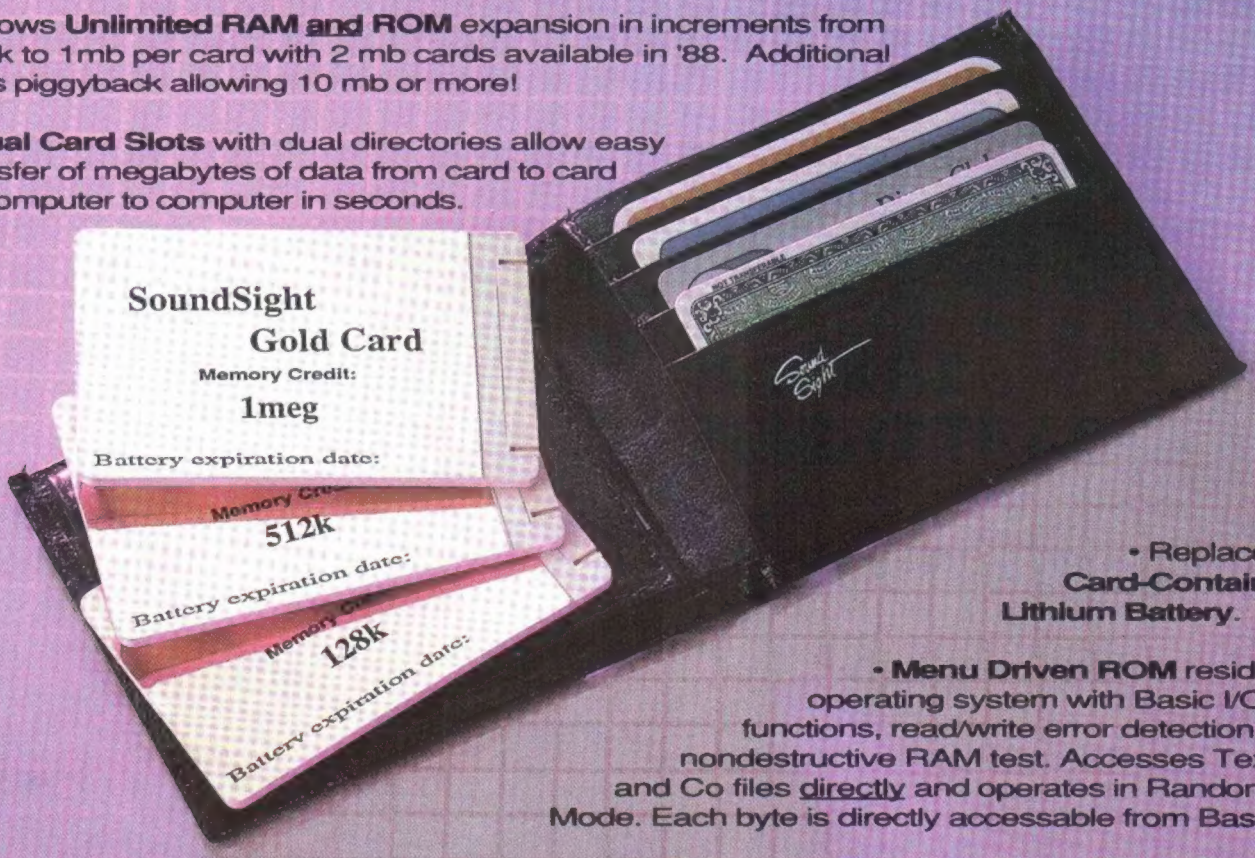
The SOUNDSIGHT GOLD CARD

Megabytes of memory on a credit-card-sized wafer
A mere 1/8" thick

*Give your **Tandy 100/102** the power it deserves without sacrificing portability or adding bulky boxes. The SoundSight Gold Card lets you add as many megabytes of **RAM and ROM** as you wish with advantages other expansions just can't match.*

• Allows **Unlimited RAM and ROM** expansion in increments from 128k to 1mb per card with 2 mb cards available in '88. Additional slots piggyback allowing 10 mb or more!

• **Dual Card Slots** with dual directories allow easy transfer of megabytes of data from card to card or computer to computer in seconds.



• Replaceable, **Card-Contained Lithium Battery**.

• **Menu Driven ROM** resident operating system with Basic I/O functions, read/write error detection, and nondestructive RAM test. Accesses Text, Basic and Co files directly and operates in Random Access Mode. Each byte is directly accessable from Basic.

• SoundSight Gold Cards are also available in permanent **One-Time-Programmable or Read-Only-Reprogrammable** memory cards for large nonvolatile software application use in conjuction with RAM cards.

Optional new **ROM ELIMINATOR** lets you load and run multiple ROMs directly from the SoundSight Gold Card, dispensing with the need for extra ROM sockets or even having your ROM chips with the computer!

NEW OPTIONAL SOFTWARE DEVELOPMENTS:

- ROM resident **Forms Generator** with full Telecommunications and Desk Top package for Law Enforcement Incident Reports, or any custom data entry applications.
- Point of Sale **Inventory Control Software**
- New **Basic Compiler** compiles basic programs to machine code, condensing them to fraction of there original size and speed. Runs them directly from cards. *(Avail. Feb. '88.)*
- New **Text Editor** opens and edits files as large as available ram. *(Avail. Feb. '88.)*

*SoundSight offers **customized hardware/software** systems for law enforcement, retail, warehousing/distribution and other vertical markets, as well as third party dev. (C language).*
We're ready to meet your needs TODAY!

Pricing available from **128k** to **10mb** with **128k at $399.95**; **256k at $549.95**; and **512k at $749.95**; Additional cards for backup are available.

CONTACT:

SoundSight MBM Inc., 225 West Broadway, Suite #509, Glendale, CA 91204
Phone (213) 463-9457 or (818) 240-8400

FREE ISSUE

Add POWER to your portable with PICO—The Magazine of PORTABLE COMPUTING !

Take your first issue FREE with the coupon below!

"Portable, Briefcase, Laptop, Notebook."

Whatever you call them, PICO sizes them all up every month. PICO is where you'll find the hardware, the software, the applications, peripherals and more that help you get **all** the power your portable can deliver!

Issue after issue, PICO's computing pros bring you valuable how-to's, tips and techniques, extended tutorials, on-target news, reviews and previews, software updates, buyer's guide – all designed to make you a better computerist, whatever your favorite portable brand or flavor.

"The hottest magazine for the hottest new technology in computing!" says tech consultant Greg Resker.

SEE FOR YOURSELF WITH YOUR FIRST ISSUE, FREE!

Complete the coupon and mail today to get your first issue of PICO FREE and without obligation of any kind. If you're not satisfied, just write "cancel" on the invoice and owe nothing. The issue is yours to keep. Otherwise, a full year of PICO's money-saving help and expertise is just $29.97, GUARANTEED. Cancel at any time and receive prompt refund of the unused portion of your subscription.

SPECIAL **BONUS**

Send payment with your order – with full refund guarantee – and get PICO's exclusive Buyer's Guide to 88 current portables including data on 17 performance catagories!

Subscription includes FREE MEMBERSHIP on *The Source* database (a $49.95 value) plus access to the PICO BBS!

Buy smart – Computer smart with PICO.
Complete the coupon now and mail today!

Mail to: PICO P.O. Box 428, Peterborough, NH 03458-0428

YES! send my first issue of PICO to examine for 30 days, FREE!
Also enter my no-risk subscription for one year at $29.97* – 30% off the cover price. Subscription includes membership in THE SOURCE, a $49.95 value, plus access to the PICO BBS. If not satisfied, I'll mark "cancel" on the invoice and owe nothing.

- ❑ Payment/Charge enclosed. Please send BONUS
- ❑ Bill me. I'll skip the BONUS—Portable Computer Buyer's Guide.
- ❑ 2 Yrs, $55 (**Save $5**) (Full refund guarantee.)
- ❑ CHARGE MY: ❑ MC ❑ Visa Card# _____ Exp. Dt. _____

(print) NAME _____

Address _____

City _____ State _____ Zip _____

Computer Type _____

*Canada & Mexico, $32.97, all other foreign, $49.97. Airmail add $50.00, US funds drawn on US banks only. Please allow 4-6 weeks for delivery of your first issue.

ON THE COVER: Nelson Bohall's space fleet, ready for launching. Photo by Nelson Bohall.

VOL. 5, NO.7
JULY 1988

ROM WITH A VIEW

SUMMER'S HERE! This is our annual combined JULY/AUGUST issue. The next issue you will be receiving will be the September issue. We combine our summer issue to take a well deserved vacation. We've had to work like one-armed paper hangers to keep up with the growth of the Notebook Computer industry.

Great News on the way... Mike Nugent is joining our editorial staff as of June 1. He's a Notebook Guru and will be responsible for technical support and the magazine's editorial content. He will also be monitoring our PBBS bulletin board. Look for some real winning tip, hints, and help in upcoming issues. Mike is a long-time user of Notebooks and has written several commercially available programs. He has also been responsible for several discoveries about the Model 100 line which will be discussed in future issues.

Lately, we've seen the Notebook computer move into some interesting new areas. This month, we have an article on how the computer is being used with model rockets. The Model 100 and its descendants seem to be pervading every area; business, hobbies, games, and more. As I have maintained in the past, these computers enhance almost every activity they touch.

Toolbox

Manuscripts were typed into Microsoft Word 3.0 on a Tandy 1400 LT (dual drive, 640K), where they were edited, spell-checked, and had basic format instructions inserted. From there they were loaded into a Tandy 4000 (80386 CPU, Tandy EGA Monitor, Tandy LP-1000 LaserPrinter) desktop computer and placed into Aldus' IBM PageMaker 2.0a. There they were dummied into a rough approximation of the magazine's final appearance. Here, pull quotes are placed, headlines, intros, and bylines are sized and positioned, and advertisements are allocated space.

Next, the magazine (divided into sections) was ported over to Diana Shonk's Macintosh Plus, using the 1400 LT and Mac-link. Diana then went over the publication using Aldus Macintosh PageMaker 2.0a, page-by-page, making final design decisions on photo, figure, and listing sizes and placements. She precisely placed the text and added all the little things that go into making a nice looking publication.

Approximate page previews were output from her Apple LaserWriter Plus. When everyone was satisfied with the appearance, the Macintosh disk was sent to Colorite Corp. in Wisconsin for final output directly onto film (and in some cases, photographic paper). The film was then delivered to the printer, who printed it, labeled it, and mailed it to you.

CHIEF EXECUTIVE OFFICER
Marlene Butland

PRESIDENT/PUBLISHER
Terry Kepner

EXECUTIVE VICE PRESIDENT
Mark Robinson

EDITOR-IN-CHIEF
Terry Kepner

EDITORS
JoAnn Niemela, Dave Klein
Linda Tiernan

CIRCULATION MANAGER
Shauna Crowley

CIRCULATION ASSISTANT
Elizabeth Klein

ADVERTISING DIRECTOR
Randy Byers

ART DIRECTION and DESKTOP PUBLISHING
Diana Shonk
ACROSS THE BOARD
Graphic Design, Inc.
8 Windy Row
West Peterborough, NH 03468-0091

PORTABLE COMPUTING INTERNATIONAL CORPORATION
145 Grove St. Ext., #21, PO Box 428
Peterborough, NH 03458-0428

Editorial 603-924-7949

Advertising 603-924-9455

Circulation 603-924-9455

Portable 100 (ISSN 0893-9420) is published by Portable Computing International Corporation, 145 Grove Street Ext., #21, P.O. Box 428, Peterborough, NH 03458-0428. *Portable 100* is an independent journal not connected with any hardware, software, or peripheral equipment manufacturer. *Portable 100* is published monthly, except for a combined July/August issue in the summer.

Subscription Service: All subscription correspondence should be addressed to *Portable 100*, Portable Computing International Corporation, 145 Grove Street Ext. #21, P.O. Box 428, Peterborough, NH 03458-0428. U.S. subscription rates: $24.97, one year; $53 two years. Canada and Mexico: US$29.97, one year; US$61 two years. All other foreign (surface mail): US$44.97, one year; US$85 two years. Foreign Air Mail, add US$50 per subscription year. All payment, except Canada, U.S. funds drawn on U.S. Bank. Second-class postage paid at Peterborough, NH 03458, and at additional mailing offices.

POSTMASTER: Send address changes to:
Portable 100, Portable Computing International Corporation, 145 Grove Street Ext., #21, P.O. Box 428, Peterborough, NH 03458-0428.

Is There a Lemon in the Slots?

TROUBLE IN SLOTS CITY

"The Slot Machine Game" in the May 1988 issue is a lengthy program—more than 200 lines requiring 7,600 bytes—that took four hours to enter and check. Running it corrupted my 100 when it was impossible to fix a "text ill-formed press space bar" message. The complete display as shown in Table 1 on page 10 never did appear, showing only an error message at 610 and a partial display.

I duplicated your published program not once, but twice, line for line, space for space.

Of course, it contains an error!

Ralph Sherman did indeed devise an interesting, even educational, slot machine program. I would, in fact, still be interested in purchasing such a program for a responsible price, say $19.95.

There seems to be an open forum about what Portable 100 should and should not be in the future.

Objective product reviews are a must. How else will 100 owners know whether something on the market will benefit them? In the last two years I have spent about $500 on products reviewed and/or advertised in Portable 100.

John Bozman
Salisbury, MD

"THE "$$###" SLOTS

I found Ralph Sherman's article "The Slot Machine Game" both enlightening and interesting, so much so that as soon as I finished reading it, I typed it into my Model 100. I did find some typos—line 650 should read *USING"$$###"* not *USING"$$#R#"* and the line following 1315 should be 1320 not {20. Also, when, bleary eyed, I tried to run the program, I got a /0 error in line 610. Checking, I found that *(W1-B)* equaled zero, so I eliminated */ ABS(W1-B)* from line 610 and found the program ran (or seemed to run) beautifully without it.

After playing the "slot" for half an hour or so (with a great thrill when it rang up three diamonds for a $100 payout), I got the feeling that something wasn't right. After checking lines 2010, 2020, and 2030 (characters for reels 1, 2, and 3), I could picture a headline in tomorrow's papers: "Portable 100 Pushes Crooked Slots". Why? Because the chart says

Portable 100 Pushes Crooked Slots

that the $ on reel three is wild and promises a $100 payout for $-$-$, but there is no $ (decimal 36) on reel three, so you can't get either one!

Further, if Sherman's program (and it's a great one) is the same as that on the real Twenty-One Bell machines, then Las Vegas and Atlantic City are full of crooked slots.

Incidentally, although I can do a fairly decent job of programming in *BASIC*, and could follow this one, I didn't understand the purpose of *CALL16959* in line 20. Perhaps someone out there can explain it for me.

Bertram I. Strauss
New City, NY

SOLUTION FOR THE SLOT

I received my May issue of Portable 100 and started typing in *SLOT.BA* as I am always trying to find more games to play on my TANDY 200. After typing it in and then trying to run the program I found it would not run on my machine. After checking what I had typed and correcting my usual errors I found it still wouldn't work. The following are some of the things I did to finally make it work for the most part. Still some problems but they might be related to the Tandy 200.

1. In line 610 they have the formula *(B1-B)/ABS(W1-B)*. I changed it to *(B1-B)/ABS(B1-B)*. This seemed to make the program work some. W1 is only listed this one place so it has nothing to compare it to.

2. I thought you meant line 1320 where you have {20.

3. In line 650 you have *$$#R#* and I typed *$$###* and that helped that.

I still have a problem with a bunch of 00000's after the 20 in the *NOW* dollars that I start with, as well as initially saying I have 20 plays. When you hit *ENTER* for the first pull, the *20 PLAYS* then says 1 play but the bunch of 0000's are still after the *NOW* amount. If you complete the game, win or lose all of your money, and then hit *ENTER* to restart, the game all resets, and everything reads right. Maybe some small problem left. Hope you can help me with this program as I really like it.

I have had my 200 for about a year and so am a novice at hacking. I had a 600 for awhile but with no support I gave it to my brother for word processing. I am learning much from your magazine and enjoy each copy. As most programs are written for the 100-102 I use the other half of the screen to add my own touches to the programs. Some of the programs can be expanded to fill the 200 screen.

Again I want to thank you for publishing the Portable 100.

Pat Galligan
Washington, D.C.

THE GAME ON TAPE?

Greetings: Love your magazine! I also have a question on "The Slot Machine Game" found in the May 1988 issue. The program is really too much for one to copy. Add to that fact the problem if one were to make just one error!! Why could you not offer that program (for a price) on a disc or cassette tape? There may even be a subscriber to your magazine who would do it for a price.

For myself I could never copy that long program correctly.

William J. Yearsley
Plymouth Meeting, PN

THE AUTHOR'S CORRECTION

I offer a deep apology to readers who typed *SLOT.BA* on their computers and found it didn't work. I spent at least 30 hours creating and testing the program, so I was disappointed to find typos in the listing.

Anyone who would like a perfect copy of *SLOT.BA* on cassette or 100K 3.5-inch floppy disk (for the old-model Tandy portable disk drive) can get a copy from me for $15, including the tape or disk, and including the shipping. I'll also throw in a correct printout of the program listing as a backup. Write to me at P.O. Box 1483, Waterbury, Conn 06721. Allow two weeks for your check to clear and another four weeks for the mail to deliver your program.

As for the specific comments in the above letters:

Mr. Strauss is correct about Line 1320 and the *PRINT USING* format in Line 650, but Line 610 should say *STEP (B1-B)/ABS(B1-B)*, as noted by reader Galligan. The correct program does not contain a variable called W1.

As for Mr. Strauss' complaint about no $ (*CHR$(36)*) on Reel 3, please see the section of my article that was subtitled "The Model 100 Simulation." In that section, I explained how some positions on some reels of the Twenty-One Bell contain two symbols, and how *SLOT.BA* covers that situation. In brief, *SLOT.BA* will give you either a $ or a 7 on Reel 3 at position 15, depending on which symbol would be more to your advantage - if there would be an advantage either way. The code for the Reel 3 decision-making is contained in Lines 1410-1435.

The instruction *CALL 16959* in Line 20 calls a built-in subroutine that keeps the display from scrolling. This instruction is necessary to maintain the display when characters are printed on the bottom line. Because of this instruction, if you run the program and press *SHIFT-BREAK*, you will be in *BASIC*, but the display won't scroll unless you type *CALL 16964* or go to the main menu or turn the computer off and on.

As for Mr. Bozman's comments, I am sure his error in Line 610 was the same as the one Mr. Strauss found, and the "text ill-formed" message indicated that there was yet another error—a syntax error so elementary that *BASIC* wouldn't accept it.

I have a Model 100 and have not run the program on a Model 200, so I can only guess at Pat Galligan's problem with the *NOW* dollar display. My guess is that Mr. or Ms. Galligan has an error in the *PRINT USING* format in Line 610. That line appeared correctly in the article, except for the aforementioned W1 variable.

Please don't give up on my articles or Portable 100 because of a few typos. With my next article, I'll submit a listing prepared on my own printer, and that should solve the problem.

Ralph Sherman
Waterbury, CT

We would like to add that we strive to make the programs as bug-free as possible, but our small staff and time deadlines somtimes make us miss a few important details. To correct that problem, we have just hired a technical editor whose job it shall be to make sure all the programs HAVE been tested before they're published. He will also be responsible for putting those programs up on our BBS on a timely schedule (before they're actually published is the goal). You should start to see the effects of his efforts starting with the September issue.

EUROPEAN VIEW?

Many users also have access to a PC and there might be more articles and product information on how to use both meaningfully in a given context. I find myself often hesitant to start the "noisy" PC and would rather use the fast and silent Model 100 but have not yet started connecting both.

Tandy-Germany has nearly totally withdrawn from the market and it might be helpful having a little section in your magazine called the "European View."

Richard Gumtau
Eichenau, W. Germany

That's an interesting idea just how could we help with such a specific column?

Eds

ANOTHER AUTHOR'S CONTRIBUTION

Enclosed is a copy of a *BASIC* program to determine starting address, number of lines, and number of bytes for any *BASIC* program on the Olivetti M10. The original version of this program was published on page 14 of the August 1987 issue of PICO. Of course, as usual, the published version had one error (Sadistic editors that make sure there is at least one major error in any program published so they can giggle when they think of the thousands of man-hours spent by their readers trying to debug the program!), and this is in statement 100: the *P$* should have been *N$*. I have also added another *PRINT* statement to statement 100 to give a better looking output on the screen.

To adapt my enclosed code for the Tandy Model 100: statement 50: change *63933* to *63930* (this is Entry 1); statement 50: change *64131* to *64128* (this is Entry 19). To adapt my enclosed code for the NEC 8201: statement 50: change *63933* to *63633* (this is Entry 1); statement 50: change *64131* to *63853* (this is Entry 21).

I have found this program to be extremely useful in controlling the way I load my portable computer to maximize the available RAM space left for document files. The file name, when requested by the program *BCNTR.BA* must be entered in capital letters with no extensions; e.g.: *BCNTR*.

Norman L. Donaldson
Granada Hills, CA

No, we're not sadistic. In fact I get quite upset everytime I find a listing with an error in it. In any case, thanks for your short program. I'm sure our other Olivetti readers will appreciate it.

Eds

HOW TO MAKE A TIMED INPUT

Gene Bordenkircher sent in a clock program that displayed messages, on certain dates, and asked how to make a timed input routine. The wonderful editors had an answer, but here is another one.

This routine uses the *INKEY$* function for the entire input module, this way it can have the 5 second time-out on the start of the message or for any character. This version has been set for any character, so that if you wait more than 5 seconds while typing, the machine will abort your message figuring you've gone to sleep, and it will continue on.

The time-out input routine is shown in listing 2. The only thing I have neglected is the cursor, *INKEY$* does not turn the cursor on, so you might want to print *ESC-P* to turn it on before this routine, and *ESC-Q* to turn it off after, or you can put your own non-blinking cursor directly into the routine. A good idea would be to remove the REM's. They are only needed to show YOU what is happening... One other suggestion to shorten the program and allow you to change the messages a bit easier. A simple change could have the program scanning a file. The file would contain the date, and the message in the form *MM/DD*, <message> you could read in each with *INPUT#1,D$:LINE INPUT#1,MM$*. The *LINE INPUT* command will allow the message itself to contain commas. This will considerably shorten the clock program and make those nifty messages a bit easier to change!

Hope this is of some help.

C. Arthur Martin Zimmerman
Calgary, AB

We appreciate your "improvements" Keep up the good work.

Eds.

```
10 REM BCNTR.BA BASIC FILE COUNTER 01/30/87 Ray N.
Herring - Revised for Olivetti M10. 04/20/88 Norm
an L. Donaldson
20 P$="":GOTO40
30 B=INT(B/16)*4096+(BMOD16)*256:RETURN
40 INPUT"FILE NAME";N$:LN=LEN(N$)
50 FORI=63933!TO64131!STEP11
60 IFPEEK(I)<>128THEN90
70 FORK=I+3TOI+2+LN:P$=P$+CHR$(PEEK(K)):NEXTK
80 IFN$=P$THEN100ELSEP$=""
90 NEXTI
100 PRINT:PRINT"PROGRAM NAME IS...";N$
110 A=PEEK(I+1):B=PEEK(I+2):GOSUB30
120 NB=B+A:NL=0:A=NB
130 B=PEEK(A+1):GOSUB30:AN=B+PEEK(A):IFAN=0THEN150
140 NL=NL+1:A=AN:GOTO130
150 PRINT"STARTING ADDRESS..";NB
160 PRINT"NUMBER OF LINES...";NL
170 PRINT"NUMBER OF BYTES...";A-NB
180 END
```

Listing 1 is a Basic program to determine Starting Address on the OM-10.

ULTRASCREEN NOT SO GREAT

I would like to respond to the article by Donald Maxwell entitled *"Ultrascreen."* In this article he made some comments which one should certainly take with a grain of salt. While I basically agree with his overall review of *Ultrascreen* there are some problems with the program which contradict what he has said.

First, *Ultrascreen*, while it may work with all the built-in programs on the Model 100 it certainly does not work with all programs written for the Model 100. The author should have taken the pains to specify which programs it does not work with. I have used *Ultrascreen* for almost a year now and I have found that it does not work with *SUPER ROM*, at least it does not work with *SUPER ROM* the way that it should. For example, when using *Ultrascreen* with *SUPER ROM*'s *Lucid* spreadsheet, the screen is half blank, the function key assignments are not properly aligned beneath the function keys. Moreover, when you select *SUPER ROM* from the Model 100's main menu and then *Lucid*, from SUPER ROM's main menu, the cursor does not line up with the prompt. Furthermore, when you enter a filename in response to *Lucid*'s prompt the filename is not echoed to the screen as it should be; the screen displays nothing and you may or may not enter the file you designated. I say you may or may not enter the file you designated for I have, on occasions, entered the file but more often than not I was returned to the Model 100's main menu. This seems to be an intermittent bug and therefore does not occur on every attempt. If you try to enter a new filename at *Lucid*'s prompt you are once again returned to the Model 100's main menu.

There is another, more devastat-

```
1ØØØ REM Timed Input routine -- T=timeout in seconds, L=maximum length
1ØØ5 M$="":M=Ø:T$=TIME$:TT=T            :REM Setup, for routine
1Ø1Ø I$=INKEY$:IF INKEY$<>"" THEN 1Ø25 :REM Wait for keys to be pressed
1Ø15 IF TIME$<>T$ THEN T=T-1:T$=TIME$  :REM Use clock for REAL seconds
1Ø2Ø IF T=Ø THEN M$="":RETURN           :REM Timeout, return empty M$
1Ø25 IF I$=CHR$(13) THEN RETURN         :REM Enter key ends input...
1Ø3Ø IF M>L THEN BEEP:GOTO 1Ø1Ø         :REM Past max length, complain...
1Ø35 IF I$<>CHR$(8) THEN GOTO 1Ø55      :REM The backspace key...
1Ø4Ø IF M=Ø THEN 1Ø1Ø                   :REM Nothing to BKSP, ignore it...
1Ø45 M=M-1:M$=LEFT$(M$,M)               :REM Remove the character from M$
1Ø5Ø PRINTI$;" ";I$;:GOTO 1Ø1Ø          :REM ...and from the screen
1Ø55 IF I$<" " THEN GOTO 1Ø1Ø           :REM Ignore any control characters
1Ø6Ø M$=M$+I$:M=M+1:GOTO 1Ø1Ø           :REM Add character to message
```

Listing 2. Using the INKEY$ routine for timed input.

ing problem using *Ultrascreen* with *SUPER ROM*'s spreadsheet; the cursor erases a cell's contents, be it a formula or a label. However, when the screen is updated the erased contents reappear. The cursor also prohibits one from seeing the contents of the cell the cursor is sitting on. Given these problems one cannot, realistically, use *Ultrascreen* with *SUPER ROM*'s spreadsheet for one would have a great deal of difficulty ever accomplishing anything.

If you select *Write Rom* from *SUPER ROM*'s main menu several things happen. First, only the left half of the screen is used; the right half remains blank and the bottom one-third of the screen is not used either. Second, the function key assignments do not line up under their respective keys. Third, the cursor is far too large for the screen display; it covers almost two complete filenames. However when you select a filename it will put you into the file and then the function key assignments are properly lined up. Thus far I have not had any problem with the function keys working properly.

When using *Thought* and *SUPER ROM*'s database one experiences similar problems thus making *Ultrascreen* more of a nuisance than a piece of software that one would recommend "without hesitation" to others. But at least one should get the idea that the only problem with *Ultrascreen* is that it doesn't work with *SUPER ROM*, (after all how many people own *SUPER ROM*?) I must make some other observations that contradict the reviewer's position.

Maxwell states that the speed at which *Ultrascreen* accepts characters is only about half as fast as normal but that this doesn't cause him to lose any characters typed given the size of the buffer, even though he has typed faster than what *Ultrascreen* can handle. Of this I can say only one of two things, either he is an exceedingly slow typist or his version of *Ultrascreen* is not the same as mine. When using *Ultrascreen* I consistently lose characters even though the buffer continues to output the characters it contains. To demonstrate this one you need only try generating a few line returns. If you press *RETURN* two times in rapid succession, say at the end of a paragraph after having typed at my normal typing speed, *Ultrascreen* does not seem to be able to keep up. This is especially true if one is only a mediocre typist such as myself let alone an experienced keyboard operator.

My next point concerns Maxwell's claims about the legibility of the characters using *Ultrascreen*; he claims that they "usually seem even easier to read" (with emphasis on the word easier) than normal characters. This is patently false unless one is using a magnifying glass. The characters are so small that it takes some getting used to before one could claim that they are even as legible as the larger normal characters, let alone "easier to read."

There is one further problem with *Ultrascreen*; this one Maxwell failed to mention, and that is what happens when the screen automatically shuts off. If you are using *Ultrascreen* with *SUPER ROM* and the screen goes blank after a period of no keyboard activity the results are completely unpredictable. I have had everything happen, when I switched the computer back on, from being half in the document I had been working on and being half in the *SUPER ROM* Write menu, i.e., at the same time. At other times I have had to use the *SHIFT-BREAK* keys to get unstuck from whatever happened. And last, but not least of all, I have had to use the *RESET* button. Thus far I have not had to switch the memory off but then I don't tend to use *Ultrascreen* that much anymore.

I typed this while using *Ultrascreen*. At first I was going to leave the missed characters, which Maxwell said he didn't suffer, in the document but there were so many it would have made it difficult to read. Thus I edited the document but nevertheless I believe there is sufficient evidence to indict Maxwell for his poor review of an otherwise unremarkable product.

Kenn Cust
Edmonton, Alberta

Thanks for sharing with us your experiences with Ultrscreen.

MORE CODES TO THE GEMINI

I read with interest in your April edition ("I/O") Lorrain Jensen's discussion of embedded printer codes, especially since I also use the Gemini 10x model. All of us who have tried using the *CTRL-P*...etc. routines, attempting to get the printer to respond correctly, have been frustrated when the Tandy 100 strips them out in certain modes.

There is, however, another set of codes which surprisingly are not stripped out by the *SHIFT-PRINT* routine, and I've included some which work just fine, although most people agree that the "ultimate solution" to problems like this is a full-featured word processing utility.

George W. Flanders
Edina, MN
Eds

Printer Cmd	Keyboard Input
Form feed	[GRPH ']
Compressed on	[GRPH e]
Compressed off	[GRPH ;]
Expanded on	[GRPH i]
Expanded off	[GRPH w]
Italic on	[GRPH k][4]
Italic off	[GRPH k][5]
Double strike	[GRPH k][6]
" off	[GRPH k][H]
Emphasized on	[GRPH k][E]
Empasized off	[GRPH k][F]
Roll	[GRPH t]
Carriage return	[GRPH =]
LPT off line	[GRPH q]
LPT on line	[GRPH u]
Printer Cmd	**Keyboard Input**
Proportional	[GRPH k][p1]
" off	[GRPH k][p0]
Superscript on	[GRPH k][S0]
Subscript on	[GRPH k][S1]
Super/sub off	[GRPH k][T]

Listing 3. The last five lines work with an Epson, but not with the Gemini.

FLY THE MAGNETIC SKIES

Thought you and your readers would like to know. I've recently learned that the airlines more and more are switching to magnetic tray tables. So if you're flying, and in doubt, use your lap for your laptop.

Stephan Lankton
Gulf Breeze, FL

❑

Laptops Shoot for the Stars

A BASIC program that predicts how high your model rocket will fly.

by Nelson E. Bohall

You've spent weeks building and polishing your pride and joy model rocket. Now, which engine do you use? *ROCKET.BA* accepts raw information regarding your model, adds the rocket engine attributes, then displays the speed and altitude it can reach. If you like model rocketry, you'll find this program invaluable because it can also optimize performance versus cost, *and it's portable.*

USING THE PROGRAM

Before accessing *ROCKET.BA*, you should make careful measurements of the bare rocket. It's ready to fly—but without an engine. So you'll weigh it *as is*. You should use a postage scale, because your rocket will weigh much less than one pound. If your model has a payload compartment, then you must also add the payload to the weighing-in. Next, measure the largest outside rocket body diameter. You may have a stepped-shell design like a Marauder, larger in the front and smaller toward the rear. This will help determine the drag coefficient.

The third entry is an estimate of the aerodynamic drag force. Try about .5 for a start. If your finished rocket flies higher than this program suggests, the drag force may be too small; if lower, try a higher value.

After these three have been entered, a menu is displayed of the nine most-used engines, plus a keyboard input. The nine standards have their characteristics loaded in *DATA* statements, so they are directly available from the menu. These attributes include total engine impulse, thrust duration (burn time), initial engine weight, and propellant weight.

The Estes Industries catalog describes their *Comet* model rocket as "1.80 oz. weight, 1.325-inch diameter and capable of reaching 1,200 feet." So when you enter the weight and diameter into the program and choose Estes' recommended C6 Engine, the computer tells you that the apogee (maximum height) is 1,051 feet with a drag coefficient of one-third. (see page 10 for *APOGEE.BA*, an altitude-tracking program to verify your earlier prediction.) So if your new model only reaches 1,051 feet, you know that to reach 1,200, you must reduce drag.

Now here is where this program shines. Without re-entering the model characteristics, you may press *E* for a new engine and try a different size. You can do this over and over until you're sure of your choice. So without spending a dime, you can make your model fly at ever increasing speeds and altitudes.

THE PROGRAM

I like clean, crisp menu-driven programs. So I'll place all of the menu titles in *DATA* statements, *READ* them out with *FOR-NEXT* loops, tidy-up the *PRINT* format, and include *LINEINPUT* statements in the loop. I departed slightly from the norm because I wanted a *units* descriptor at each input query. For example, the weight should be in ounces, diameter in inches, no units for the drag, lb-secs for the impulse, and so on.

Following the first three entries, line 19 headlines the N.A.R. engine types. Again *READ* from *DATA* to list on the screen the nine types. The first is a 1/2 A6, so print the *1/2* using *CHR$(174)* at line 21 before the list is displayed.

All the *LINEINPUT*s are loaded as *I$(I)*s; that is, *I$(4)*, for example is the *engine impulse* in lb-secs, while *I$(7)* is the *propellant weight* in ounces. Without this clue, it may be difficult to follow the program flow. The output functions are likewise strobed with *FOR-NEXT*s and stored as *O(I)*s starting at line 29. Here is a list the variables by name and function:

RV$ Reverse video
VR$ Video reverse (black-on white)
CD$ Clear display (as distinct from CLS)
PI The value of PI (3.1415926)
I(1) Rocket weight
I(2) Rocket diameter
I(3) Drag force

I(4)	Engine impulse
I(5)	Thrust duration
I(6)	Engine weight
I(7)	Propellant weight
CS	Rocket cross-section area
W0	Initial weight at launch
W1	Average weight during burn
W2	Coasting weight after burn
AC	Acceleration during burn
C1	Air foil coefficient during acceleration
C2	Air foil coefficient after burn
TA	Average engine thrust of engine
O(1)	Burnout altitude
O(2)	Burnout velocity
O(3)	Coasting distance
O(4)	Maximum altitude
O(5)	Coasting time

The calculations in lines 29-32 were developed by Douglas J. Malewicki, an aeronautical engineer in Phoenix, AZ. They include factors such as air temperature and density, plus the air resistance exponentially increasing from acceleration.

These may not be vital to rockets that cannot go faster than 1,000 feet per second. Still, 1,000 feet per second is an unbelievable 681 MPH or MACH 0.9! Most of our models peak at less that 500 ft/sec (this is still 340 MPH).

If there are model rocket readers who'd like this and the APOGEE.BA programs on disk or tape, you may write me and include $10.00 for postage, disk, and handling. Send to Nelson E. Bohal, 1518 9th Ave North, Edmonds Washington, 98020-2627. ❑

```
10 'Rocket.BA Nelson E Bohall, 7/85:Rev
Ø2/Ø5/88:Rev Ø3/2Ø/88
11 ES$=CHR$(27):RV$=ES$+"p":VR$=ES$+"q":
CD$=ES$+"J"
12 DIM RE(9,4) :PI=4*ATN(1) :RA=18Ø/PI :
U$="##.###" :V$="####" :GOTO 15
13 Q$=INPUT$(1):IF Q$>"Z"THENQ$=CHR$(ASC
(Q$)-32):RETURN ELSE RETURN
14 LINEINPUTI$(I):I(I)=VAL(I$(I)):RETURN
15 CLS:PRINT@6,RV$" ROCKET FLIGHT PREDIC
TIONS "VR$
16 RESTORE41:FORI=1 TO 7:READH$(I),E$(I)
:NEXT
17 FORI=1TO3:PRINT@44+I*4Ø,H$(I)STRING$(
2Ø-LEN(H$(I)),241)TAB(31)E$(I):NEXT
18 FORI=1TO3:PRINT@65+I*4Ø,"";:GOSUB14:P
RINT@64+I*4Ø,USINGU$;I(I):NEXT
19 RESTORE48:FORI=1TO9:FORJ=1TO4:READRE(
I,J):NEXTJ,I
2Ø RESTORE47:PRINT@29Ø,"N.A.R. Engine Ty
pes":PRINT:FORI=ØTO9:READRE$(I):L(I)=LEN
(RE$(I)):NEXT
21 FORI=ØTO9STEP2:IFI=ØTHENPRINTTAB(9)CH
R$(174);
22 PRINTTAB(11)RE$(I)STRING$(5-L(I),241)
;I;TAB(2Ø);I+1;STRING$(5-L(I),241);RE$(I
+1):NEXT:GOSUB13
23 PRINT@244,CD$:FORI=4TO7:PRINT@44+I*4Ø
,H$(I)STRING$(2Ø-LEN(H$(I)),241)TAB(31)E
$(I):NEXT:Q=VAL(Q$)+1:IFQ=1ØTHEN25
24 FORI=1TO4 :PRINT@64+(I+3)*4Ø,USINGU$;
 RE(Q,I) :I(I+3)=RE(Q,I):NEXT:GOTO26
25 FORI=4TO7:PRINT@66+I*4Ø,"";:GOSUB14 :
PRINT@64+I*4Ø,USINGU$;I(I):NEXT
26 PRINT@371,RV$" Engine Type "RE$(Q-1)V
R$
27 CS=PI*I(2)^2/4*(I(3)+1E-6) :W2=I(6)+I
(1)-I(7)/2 :WØ=I(1)+I(6)+I(7):W1=WØ-I(7)
/2 :W2=I(1)+I(6)
28 TA=I(4)*16/I(5) :A1=(TA/W1)-1 :C1=W1/
CS :C2=W2/CS :X=.36981*I(5)*SQR(A1/C1)
29 O(1)=235.26*C1*LOG((EXP(X)+EXP(-X))/2
)
3Ø O(2)=87*SQR(A1*C1)*(EXP(X)-EXP(-X))/(
EXP(X)+EXP(-X))
31 O(3)=117.63*C2*LOG(1+O(2)^2/(C2*7569.
386))
32 O(4)=O(1)+O(3) :O(5)=2.7Ø14*SQR(C2)*A
TN(O(2)/(87*SQR(C2)))
33 RESTORE 45:FORI=1TO5:READ O$(I),T$(I)
34 PRINT@364+I*4Ø,O$(I)TAB(24);USINGV$;O
(I);:PRINTTAB(3Ø)T$(I):NEXT
35 FOR M=1 TO 5ØØ:NEXT
36 PRINT@6Ø1,RV$" (A)nother (E)ngine (F)
light (Q)uit "VR$;:GOSUB 13:Q=VAL(Q$):ON
INSTR("AEQF",Q$)GOTO4Ø,39,38,51
38 MENU
39 PRINT@2ØØ,CD$:GOTO2Ø
4Ø PRINT@8Ø,CD$:GOTO 16
41 DATARocket Weight ,ounces,Rocket Diam
eter ,inches
42 DATADrag Coefficient ,no units, Engin
e Impulse
43 DATAlb-sec,Thrust Duration ,seconds,E
ngine Weight
44 DATAounces,Propellant Weight ,ounces
45 DATABurnout Altitude ,feet,Burnout Ve
locity,ft/sec,Coasting Distance
46 DATAfeet,Maximum Altitude,feet, Coast
ing Time,seconds
47 DATAA6 ," A8",B4 ," B6",B8 ," C5",C6
," D11",D12 ," Keybd"
48 DATA .28,.2Ø,.53,.Ø55,.56,.32,.57,.11
,1.12,1.2,.7Ø,.294
49 DATA 1.12,.83,.68,.22,1.12,.6Ø,.68,.2
2,2.25,2.1Ø,.9Ø,.45
5Ø DATA 2.25,1.7Ø,.91,.44,3.37,1.4,1.51,
.65,4.48,1.7,1.49,.879:
51 RUN "FLIGHT.BA"
```

A program to assist you in selecting the right engine for your model rocket.

Rocket Flight Tracking

After launching your rocket, how do you verify how high it went?

by Nelson E. Bohall

It's a clear, warm and windless day for the spring model rocket meet. Perhaps two dozen young rocketeers and their mentors ready models for some spectacular launches. Today's contest will have some maximum altitude and maximum glide-time winners. But for an accurate measurement of the maximum altitude, we'll need more than subjective viewers to judge these events.

LAYING OUT THE TRACKING STATIONS

We start with two ALTITRAK hand-held sighting devices. If it's a small meet and an absolutely windless day, we can get away with only one, but these conditions are rare. Figure 2 illustrates the path of a typical rocket model on a slightly windy day showing the triangle problem we'll solve. Note that its *apogee*, or maximum altitude, is not directly above the launch site.

This infers that measurements taken with only one tracking device will be in error by the wind shift in degrees. This error can be minimized by using two trackers instead of one. To eliminate completely this residual error, we should use two *theodolites*—surveying instruments—even homemade ones, which provide both elevation and azimuth readings.

ENTER *APOGEE.BA*

When we run *APOGEE.BA*, we'll see six entry points for input. These are shown in the *APOGEE.BA* menu (see Figure 1a). As a rule of thumb, we'll space tracking stations no less than one-fourth the expected altitude of the rockets as the distance from the launch site. In Figure 1b (a screen print of a flight), notice that the trackers are an equal three hundred feet from the launch site in a straight line with the launching spot for the rocket.

If we have only the hand-held tracking devices, we'll enter the *Elevation* from each of them and put a zero (or *ENTER*) in the *Azimuth* slot. If the wind is mild, our error will be fairly small.

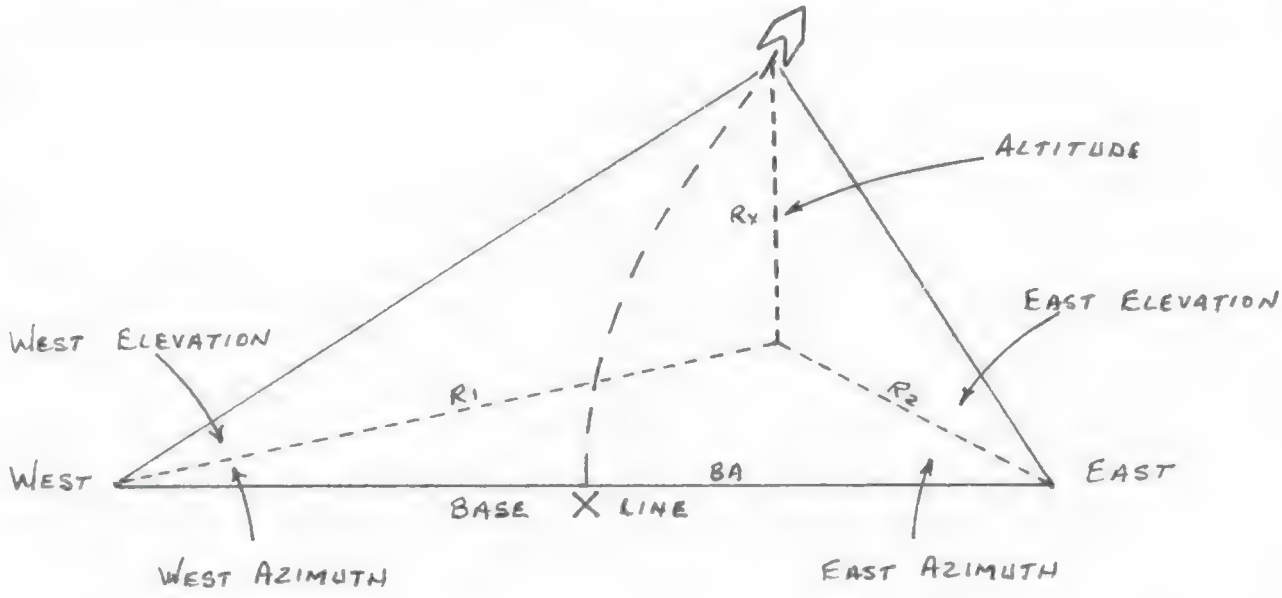
If wind is blowing from west to east, the rocket will drift away from one tracker and toward the other. As you can see in Figure 2, the viewed angle of each tracker will be different.

```
        Flight Tracking

West Tracker Distance      feet
East Tracker Distance      feet
West Elevation Angle       °
West Azimuth Angle         °
East Elevation Angle       °
East Azimuth Angle         °


Rocket Apogee is       825 feet
```

Figure 1a (top). The APOGEE.BA Menu. Figure 1b (bottom). Flight tracking results from APOGEE.BA



$$R_1 = \frac{SIN(WA) * BA}{SIN(WA + EA)}$$

$$R_2 = \frac{SIN(EA) * BA}{SIN(WA + EA)}$$

$$BB = R_1 + R_2$$

$$R_X = \frac{BB * SIN(WE) * SIN(EE)}{SIN(WE + EE)}$$

Figure 2. A model rocket's typical flight path, and how to triangulate its position. The equations used to derive its actual altitude are also shown.

```
10 'APOGEE.BA Nelson E Bohall, 7/85:Rev
02/05/88 :Rev 03/21/88
11 ES$=CHR$(27):RV$=ES$+"p":VR$=ES$+"q"
:CD$=ES$+"J":RA=57.2957795 :GOTO 14
12 Q$=INPUT$(1):IF Q$>"Z" THEN Q$=CHR$(A
SC(Q$)-32):RETURN ELSE RETURN
13 LINEINPUTI$(I):I(I)=1E-4+VAL(I$(I))/R
A:RETURN
14 CLS :GOSUB 27 :RESTORE 25
15 FOR I=1 TO 6:READH$(I):T$="feet":IF I
>2 THEN T$=CHR$(166)
16 PRINT@84+I*40,H$(I)TAB(30)T$:NEXT
17 FOR I=1 TO 2:PRINT@106+I*40,"";:GOSU
B 13:NEXT:GOTO 24
18 BL=(I(1)+I(2))*RA :CC=BL/(SIN(I(4)+I(
6)))
19 RX=CC*(SIN(I(4))+SIN(I(6)))*SIN(I(3))
*SIN(I(5))/SIN(I(3)+I(5))
20 PRINT@404,"Rocket Apogee is"TAB(25)US
ING"####";RX;:PRINT" feet"
21 PRINT@601,RV$" (A)nother, (R)ocket, (
N)ew, (Q)uit "VR$;:GOSUB12:ON INSTR("ARQ
N",Q$) GOTO 23,28,29,14: GOTO21
22 GOSUB 27:FOR I=1 TO 2 :PRINT@84+I*40,
 H$(I)TAB(25);I(I)TAB(30)"feet" :NEXT
23 PRINT@240,CD$ :FOR I=3 TO 6:PRINT@84+
I*40,H$(I);TAB(30) CHR$(166) :NEXT
24 FOR I=3 TO 6:PRINT@107+I*40,"";:GOSU
B 13:NEXT:GOTO18
25 DATAWest Tracker Distance,East Tracke
r Distance ,West Elevation Angle
26 DATAWest Azimuth Angle,East Elevation
 Angle,East Azimuth Angle
27 PRINT@51,RV$" Flight Tracking "VR$ :R
ETURN
28 RUN "ROCKET.BA"
29 MENU
```

Listing 1. The program APOGEE.BA, used to determine the altitude attained by your model rocket.

The *APOGEE.BA* Program

I like often-used constants and subroutines near the front of the program. Line 11 contains the escape functions, *ES$*, *RV$*, *VR$*, *CD$*, and *RA*, which provide escape, reverse video, normal video, clear display and the radian constant.

The menu for the six entries is stored in *DATA* statements. I strobe these out of data via *FOR-NEXT*, *READ*, and then *PRINT* these items. This provides a blank menu as seen in Figure 1a. Going back to the first entry line, I *LINEINPUT* the first two items, the tracking station distances, then *GOSUB* to line 24. Here we pick up the tracking angles for each flight. Because these values may be used over and over, we can start a new loop to include the angles our trackers report on. Note that in line 13, the *LINEINPUT*s provide conversion from degrees to radians and add the factor 1E-4. This prevents a division-by-zero error in lines 18 and 19 when zero degrees are entered. You can follow the program logic easily knowing the functions involved:

I(1) and I(2)	west and east tracker distances
I(3) and I(5)	west and east elevations
I(4) and I(6)	west and east azimuths
BA and RX	refer to Figure 2

After all six inputs have been entered, the apogee is displayed, followed by another menu. Choosing *A* sends the program routing to line 23 where the previous angle displays are erased and we add new values. The routing then folds back to line 18 for recalculation. Choosing *N* sends us back to the beginning for new baseline settings. *R* calls the *ROCKET.BA* program for new model predictions.

This program should be paired with the *ROCKET.BA* program. ❑

References

Centuri Engineering Co.
P. O. Box 1988
Phoenix, AZ 85001

Estes Industries, Inc.
P. O. Box 227
1295 H Street
Penrose, CO 81240

Handbook of Model Rocketry, Fifth Edition, G. Harry Stine

Model Rocketry, Peter Lowry and Field Griffith

National Association of Rocketry
P. O. Box 725
New Providence, NJ 07974

Technical Reports

From Centuri Engineering Co.:
TIR 30 Stability of Rockets in Flight
TIR 33 Calculating CP of Model Rockets
TIR 100 Model Rocket Altitude Performance

From Estes Industries, Inc.:
TR 11 Aerodynamic Drag Cat. # 2843, $1.80
TR 10 Model Rocket Altitude Predictions, Cat. # 2842, $1.80
Handbook of Model Rocketry by G. Harry Stine, Cat. # 2860, $10.95

Quick, Easy and Cheap Model 100 to MS-DOS Computer File Transfers

A simple batch processing file that simplifies file transfers without expensive cables or software.

by Beverly Howard

Sometimes the quickest way between two points is not the high speed route. Compare, for example, traveling between two cities about fifty miles apart by either jetliner (600 mph) or car (55 mph). The jetliner travels faster, but the time necessary to get to and from each airport, navigating the concourses, purchasing tickets, getting the plane to and from the runways, climb, descent and perhaps holding time, all add up to make the slowpoke automobile the better choice.

In the case of transferring text from a Model 100, you can meet with the same kind of delays, loading and executing telecommunications or transfer programs that can run at 9600 baud or better. The serial ports on any MS-DOS or IBM desktop computer can be accessed directly from DOS, and a couple of undocumented Model 100 tricks can be turned into a file transfer system that can make your portable an effective partner to your desk-bound computing.

The commands are very easy to use and could be typed directly from the DOS prompt. But I have included them here in a batch file for ease (see listing 1). In addition, you need an available serial port, null modem adapter, serial cable, and the MS-DOS program *MODE.COM* or *MODE.EXE* available on the default disk or accessible through the MS-DOS *PATH*. Note that the serial ports on most IBM's and compatibles shipped in the past few years use the newer DB-9 serial port connectors rather than the more familiar DB-25 or RS-232 connectors, so you may also need a DB-9 TO DB-25 conversion cable in place of the serial cable.

Since this process is a series of operations out of the widely used MS-DOS operating system, and the transported file is ready for use without additional manipulation, you'll see many options open up, expanding the use of portables in a mixed computer environment. For example, using the *RUN* option in *dBase III+* will allow data remotely entered on a portable to be entered directly into a running application. With a word processor that allows *RUN* or shell commands, a secretary could dump a rough draft or a letter written by the boss on a Model 100 directly into a document in progress without exiting the word processor.

This batch file can be created by using any text editor.

The transfer process is implemented by connecting the computers with the null modem and serial cable, executing the following batch file (named *PORT-IBM.BAT*) on the IBM compatible computer, and following the on-screen instructions.

This batch file can be created by using any text editor, such as *EDLINE* or *Wordstar* in the "non-document" mode, on the IBM that will create pure ASCII files. Most word processor programs have this feature listed as either "create ASCII" or "create TEXT ONLY" options. Create the above batch file and name it *PORT-IBM.BAT*. If you are experienced with MS-DOS, you can see and understand the commands as they are executed.

The critical text in the file is shown in all caps with associated symbols and numerals. The lower-case text that follows each *ECHO* statement is simply on-screen help for the user.

After the computers are connected, type *PORT-IBM n filename* and press *ENTER* on the desktop's keyboard where *n* is the number of the MS-DOS machine's COM port, and *filename* is the name you wish to call the transferred file. Then follow the on-screen instructions.

Batch programs are part of every MS-DOS/IBM system and therefore free and universally accessible. You

could even keep a copy of the batch commands in your Model 100 and type in the commands to transfer a copy of the program to a desktop computer before your first transfer. Remember, whatever you name the program, it must have the extension *.BAT* before DOS will execute it.

There are only four MS-DOS commands in the above batch file and only two of them are critical to the transfer process. The *MODE* command needs to be implemented only once before the first transfer. Again, the lower-case *n* is the DOS COM port number. Take my word for it, using the batch file will save a lot of grief. The main commands are *MODE COMn:2400,N,7,1,p* and *COPY COMn: filename.TXT*.

Text files will be received with carriage returns only where you actually hit the *ENTER* key when typing into the file. This means that the paragraphs will probably have to be reformatted by using a "Global Reformat" command within your IBM word processor. Because of this, formatted text, such as lists or program segments, entered on your portable should be double spaced to escape an unwanted paragraph format.

There are only four MS-DOS commands in the batch file and only two of them are critical to the transfer process.

The main undocumented "trick" on the Model 100 is the *SAVE TO:* option available in *TEXT* and *BASIC's EDIT* mode by pressing the *F-3* key. Normally you would answer with a file name and the output would be routed to an attached cassette recorder, but if you answer with

```
ECHO OFF
CLS
IF 1 == %1 GOTO START
IF 2 == %1 GOTO START
IF 3 == %1 GOTO START
IF 4 == %1 GOTO START
echo         Please check your syntax and try again. The
correct syntax is;
echo.
echo "PORT-IBM n filename"
echo.
echo              where;
echo.
echo "n"          is the COM port number (1-4) and
echo.
echo "filename"   is the desired destination filename (8
characters max,
echo                 the extension ".TXT" will be added
automatically)
echo.
echo The transfer rate is 2400 baud, and the Model 100's
STAT = 67N1D
echo.
echo The received file will be written to the 'default'
drive
echo         unless you add a drive letter & colon to the
filename.
echo.
GOTO ENDBAT
:START
if exist %2.TXT GOTO SAFE
GOTO CONT
:SAFE
echo.
echo.
echo A file with the name %2.TXT is already on the disk.
echo.
echo To avoid over-writing (erasing) it,
echo    you may type a CTRL-C now to abort this program.
echo.
PAUSE
:CONT
CLS
MODE COM%1:2400,N,7,1,p
echo.
echo If there was an error message...
echo.
echo remember, MODE.COM must be available to initialize the
COM port.
echo.
echo "Illegal device name" indicates the absence or
unavailability of COM port
echo.
PAUSE
CLS
echo.
echo Ready to Receive!
echo.
echo From within the TEXT file to be transferred
echo      (or from within Basic's EDIT mode), and on the
Model 100 keyboard,
echo.
echo  press "F-3" and answer "COM:67N1D enter"
echo.
echo.
echo (If the message "Fault Reading COMn:" appears, select
"R"etry)
echo.
COPY COM%1: %2.TXT
echo.
```

continued

Listing 1. A batch file for your MS-DOS computer to facilitate Tandy Notebook to MS-DOS file transfers.

```
echo (If the message "0 files copied" appeared, type a CTRL-
C
echo         and run this file again.)
echo.
echo.
echo Press any key to 'type' the disk file received.
PAUSE
CLS
TYPE %2.TXT
GOTO ENDBAT
:ENDBAT
```

End of listing.

COM:67N1D the contents of the file will be routed to the serial port at 2400 baud. In addition, the linefeed problem will be solved as the Model 100 appears to send a linefeed rather than a carriage return in response to this command.

If only a portion of the file is transferred, note the last characters that did transfer and check the text at that location in the Model 100. You will probably find a Model 100 graphics character at that location. If there are no graphics characters in the file, check your MS-DOS CONFIG.SYS file and confirm that the statement *BUFFERS = xx* is present and the value of *xx* is at or above 20.

2400 baud is the maximum effective baud rate I have been able to achieve without losing part of a file. This is strange, since I can use PIP to transfer files to and from my CP/M Kaypro computer at 9600 baud. As mentioned, implementing and using programs that will transfer at higher speeds will usually result in longer overall transfer times because of the additional load, setup, and file processing steps.

ADDITIONAL NOTES

1. If you get the DOS error message *Fault reading COM:1*, answer *<R>etry* before pressing *F-3* on the Model 100. If the *Fault* message reappears, there is either a hardware or MODE problem.
2. The actual file transfer from the Model 100 is complete when the flashing cursor returns to the Model 100's screen. (A long file may take a couple of minutes.) If the desktop computer didn't burp and write the file to disk when the cursor reappears, check for problems like a loose or wrong connection, wrong STAT on the portable, or lack of an end-of-file marker after the transmitted file.
3. Your machine or operating software may expect an end-of-file marker. An undocumented trick is inserting this EOF marker in your Model 100 text before sending the file. To do this, go to the very end of the text and type a *GRAPH-L* which will appear as an ->. (Inserting a *CTRL-Z* with the *CTRL-P* command does NOT work.) When the operating software "sees" this character (or many of the Model 100's other graphics characters) as part of the copy process, it will close the file and write it to disk.
4. If you need to add this character and forget, you can exit to TELCOM after the transfer is complete on the portable, select *F-4* for TERM, type a *CTRL-Z*, and the file will be closed and written to disk. Relax . . . the computers are very patient and there is no need to rush any of these steps at any point of the process.

2400 baud is the maximum effective baud rate.

5. If you immediately get the message *0 files copied*, there are two possible solutions. Some stray characters may have been waiting in the serial port buffer (from typing more than one *CTRL-Z* after the last file or stumbling around on your first few tries, for example). Start over and try again, because COPY has cleared the buffer while presenting the error message.
6. Otherwise, if the message *0 files copied* appears before EVERY transfer, omit the *GRAPH-L* character at the end of the Model 100 text file. The need for this end-of-file marker seems to vary with the specific hardware and/or DOS setup.
7. Your MS-DOS buffers should be set to a minimum of 20. (See your CONFIG.SYS file or refer to your MS-DOS manual.)
8. The *SAVE TO:COM:67N1D* command will change the default *STAT* in the TELCOM program. Don't let that trip you up during your next TELCOM session.
9. The reverse of the above commands (transferring TO the Model 100) work as well using the *F-2* key (*Load From:*) and DOS copying a text file TO the COM port. The only quirk is the need to reissue the *F-2* command followed by *BREAK* to get the file to appear after it has been sent by the IBM.
10. You can ABORT this transfer operation at any time by pressing *SHIFT-BREAK* on the Model-100 and *CONTROL-SCROLL LOCK* on the MS-DOS machine.
11. If you want to get creative with the text in the batch file, avoid any use of the characters < > | \ because they affect batch file operations even though they may be in an *ECHO* line.

❑

Save Money
Re-Ink Your Pens & Ribbons

Reuse your DMP-105/106 ribbon cartridges and CGP-115 and PC-2 printer ball point pens

by Hal Boulware

Quite often I have spent $7.95 on DMP-105 ribbon cartridges that run out of ink quickly. So I tried to re-ink these cartridges with stamp pad ink. I discovered that the ink in the DMP-105 cartridge is an oil base ink while Carters Stamp Pad Ink that I use to re-ink ball point pens is water based.

I finally found an ink that was compatible—Shachihata X-Stamper Line Dater Ink, one that can be found at most office supply stores. A 10-cc/ml bottle cost me about $2.75. I have been using this same bottle to re-ink ribbon cartridges for over two years now. Where the $7.95 ribbon typically lasted two weeks, my re-ink ribbon lasts about three times as long, and I always have a nice, dark printing ribbon. Re-inking is a simple task, and it takes only a few minutes with each application.

First, remove the ink cartridge from the printer and wind the ribbon back into the cartridge with the little winding knob on top of the cartridge. Along the long edges of the cartridge, you will notice four small slots. Insert the tip of a small screwdriver into these slots, twist the screwdriver to pry up, and after prying up all four, cautiously lift up the cartridge top, being careful not to disturb the parts inside the cartridge.

You will notice a round sponge roller about 3/4 inch in diameter. Squeeze out some ink onto the top of the sponge and let the ink soak in. The ink absorbs slowly, so you may have to give it several applications. With practice, you will learn how much ink to apply without overdoing it. I usually give a minimum of two applications, sometimes three.

Press the cover back on the cartridge and put it away until needed. Now the cartridge will dry out like a stamp pad. I have ribbon cartridges that are two years old and have been re-inked over a dozen times, yet they show no signs of wearing out. It is a process that in my estimation saves you $7.90, because re-inking reduces the cost of the cartridges to about a nickel each.

RE-INKING BALL POINT PENS

Re-inking the little ball point pens used in the CGP-115 and PC-2 printers is a little more complicated and tedious but will keep you from having to buy these expensive ball point pens ever again. I haven't purchased any pens in over five years.

As with the ribbon cartridges, the tube of pens you get from Radio Shack for the CGP-115 and the PC-2 printers don't last very long and can be costly. I always enjoyed printing out color graphics with new pens to get nicely colored drawings, but I believed that I could get nicely colored drawings less expensively. So I did some experimenting with injecting them with more ink. The following process is a result of my experimentation.

First, acquire a hypodermic needle with which to inject the ink. (Being a diabetic, I had extra hypodermic needles available for this experiment.) Next, if you have one, place a micrometer to the needle of the hypodermic and get a reading on it. This will enable you to select the proper size drill bit. I used a 61-80 drill set, which worked well. Once you've located the proper drill bit, drill a hole in the little plastic-colored collar around the end of the pen. Then inject a drop or two of Carters Stamp Pad Ink into the little ball point pen. Don't worry—this ink washes off. One to two drops is plenty, but if you overdo it you can easily wipe off the excess ink. Keep a glass of water handy to clean out the hypodermic needle between colors. Carters ink is water color, so a little water left in the needle won't hurt anything.

After applying the ink, replace the plastic caps back on the pens, and store them in the tubes in which they originally came. This way you will always have a fresh re-ink in your pens. Carters has all the colors you need—black, red, green, and blue. I've even seen purple in some stores. Don't buy large bottles of ink because you don't need but a drop or two to re-ink a pen. After re-inking my pen for over four years, I still have my original bottles of ink.

SUPPLIES

You need the proper drill bit, a hand tool to drill with, or you can use a pin vise. The hypodermic needle can be purchased from a druggist. Druggists may ask you for a prescription, but they usually sell them to you without one, if you assure them you are not going to use it to take dope. The needles are reasonable in price, usually around a quarter or so. And, of course, you need the small bottles of Carters Stamp Pad Ink. ❑

Input to your Model 100 from 15 feet away with a...flashlight?

Using a photodetector, connected to the Bar Code port, to "talk" to your Tandy 100/102

by Frank W. Schrader

Even if you do not have a bar code wand, you can put the bar code input port on your Model 100 to work. The uses for the discrete input circuit described in this article are limited only by your imagination.

The construction is easy, and all the parts are inexpensive and readily available from a Radio Shack store or an electronic supply house. You don't need to use a printed circuit board, but the template and appropriate remarks for one are included for the convenience of those who enjoy making printed circuits, or who like to impress people who admire *neat*. Perforated circuit board and hook-up wire will serve just as well. Even the opto-coupler could be dispensed with, but its use is highly recommended to protect the bar code input circuitry. The opto-coupler shown in the parts list is sold in a package of three that comes with one each: photo transistor input, photo Darlington input, and one combination input. Use the photo transistor type and save the rest for future experiments. The package costs less than two dollars.

The DB-9 plug poses a bit of a problem, because readily available units are for chassis mounting and will not plug into the Model 100 socket unless you cut the brass flange to clear the computer's plastic case. The other alternative is to have your friendly Radio Shack manager order you a replacement bar code wand plug from Fort Worth.

The circuit board, computer, and typical flashlight.

If you choose to go with the etched circuit board, component placement is not critical. Just be careful that holes for the IC or an IC socket are precisely spaced at .100 inch center distance in the two rows and that the rows are precisely spaced .300 inch apart. All wire holes should be approximately 1/16-inch diameter or less. Drill a 9/32-inch diameter hole for the momentary contact switch mounting where shown. If you choose to solder the IC on the board without a socket, be careful to observe the precautions for minimizing soldering temperatures. Nothing else is critical. A piece of black shrink tubing shrunk around a shaped dowel pin makes a nice hood for the

```
0 CLS
10 IF 8 AND INP(179) THEN 100:GOTO 10 ELSE 10
100 BEEP:C=C+1:PRINT@135,C
110 IF 8 AND NOT INP(179) THEN 10:GOTO 110 ELSE 110
```

Listing 1. Program to read the input from the photodetector.

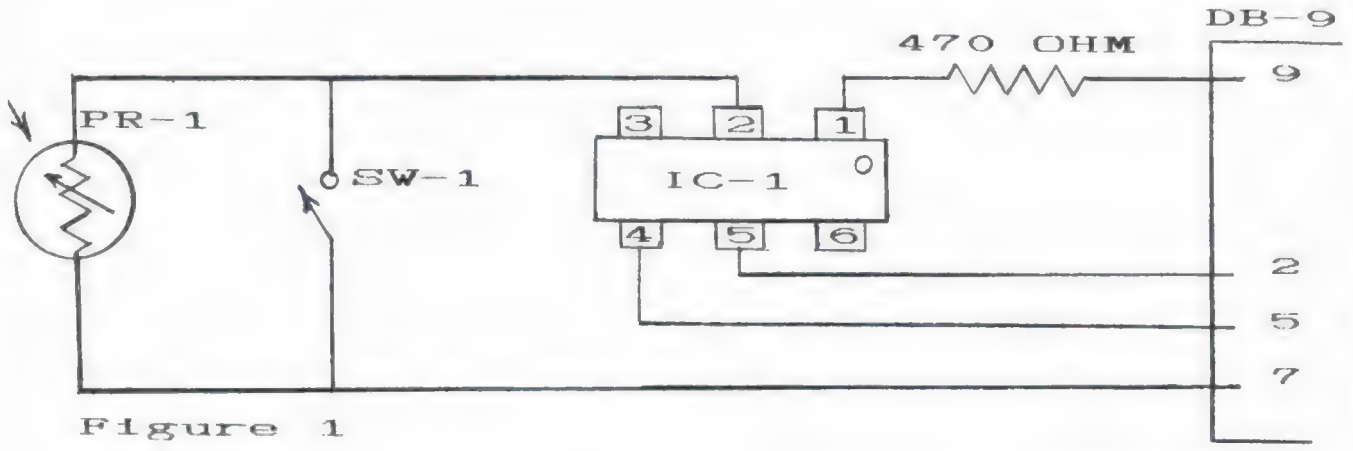


Figure 1. The schematic diagram for this project.

cadmium sulfide photo cell. This will keep out unwanted ambient light.

After assembling the device and carefully checking the circuit, you can plug it into the computer. Type in the *BASIC* program given as program Listing 1. This program is simple but will serve to illustrate the use of the device and get you thinking in terms of "Flashlight" input. Hit *F4* to execute the program. Press the momentary contact switch button. The computer should beep and show a count of *1* in the middle of the LCD. Shining a flash light beam into the photo cell WILL DO THE SAME! Beeping and counting once for each flash. Voila! We have a counting device! Changing the logic at line 10 from *AND* to *AND NOT* and at line 110, from *AND NOT* to *AND* will reverse the situation and cause the computer to beep and count when a steady beam is interrupted. With this setup, you can count coins, marbles or small children that pass through and interrupt the beam. As stated at the outset, uses are limited only by the extent of your imagination. For instance you could have a program started by the morning sun light or by the evening darkness.

Different combinations of logic statements, of course, produce different results. Experiment with the program, nothing can be harmed or cause the computer to lock up or bomb out. If you inadvertently create an endless loop, just hit *SHIFT-BREAK* to break out of the loop, or — at worst —press your computer's reset button for a warm restart to rethink your logic and fix your program.

Here's how it works: Light falling on the cadmium sulfide cell causes the resistance of the cell to drop from about .5 megohm to approximately 3 kilohm or less. This causes the light-emitting diode in the IC to illuminate the photo transistor in the IC, effectively shorting pin two of the bar code input and sending a logic one (1) to the CPU. *AND*ing this with eight evaluates to zero causing the computer to jump to another instruction; otherwise the *ELSE* statement executes and the program waits for input to continue.

The IC gets the necessary 5 volts to activate the internal light emitting diode from pin nine of the bar code socket. The 470-ohm resistor protects this supply and the diode. Values as low as 250 ohms are O.K. The diode current should be limited to the maximum shown on the package, about 20 mA. ❑

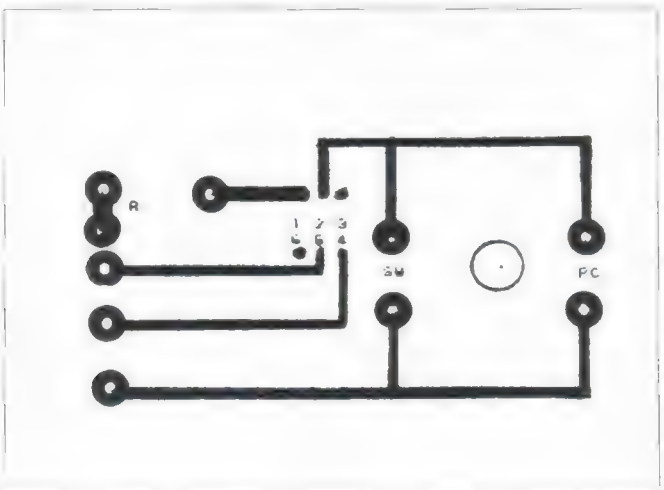


Figure 2. Printed circuit board template, bottom view, exact size.

PR-1	Photo Cell, Cadmium Sulfide, RS 276-116A or equivalent
SW-1	Contact Switch, normally open,RS 275-1547
IC-1	Optocoupler, H11AX or equivalent. See text.
DB-9	Connector, female RS 276-1538
Resistor	470 ohm, 1/4 Watt
Copper clad PC board or perforated base board.	
Misc. hook up wire.	
Shrink tubing optional	

Figure 3. Parts list.

Make your Model 100, 102, or 200 practically perfect

PCSG has long been the champion of the Model 100 and 200 laptop portable computers. No longer toys, these machines can be made to rival the performance of many desktop models.

PCSG in the past three years has developed software programs and innovations that have rewritten all standards

We can give you what we regard as the superior Model 100 or 200 system. This is a catalog of what you need to build that nearly perfect portable computer. Full brochures and the answers to any of your questions are just a quick phone call away. So, if you want to add dimensions of portable performance to your laptop unit, call in your order today.

PCSG sells every product on a 30-day full refund trial so there's no risk.

Snap-in ROMS

Super ROM

Get Lucid Spreadsheet, Write ROM, Lucid Database and Thought Outliner all on one ROM. Completely integrated so they all work together with the familiar Cut and Paste working between all applications Guaranteed to be the finest four programs available for the Model 100, 102 or 200 on one ROM $199.95, plus shipping

Write ROM

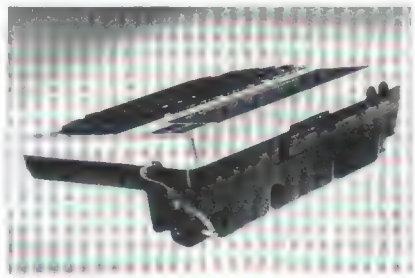
This is full word processing capability like on a desktop. Not only format documents with left and right hand margins, page numbers, headers, footers but phenomenally more. Write ROM gives you memory writing performance: 44 features and functions that really bring text processing power to the Model 100, 102. $99.95, plus shipping

Lucid ROM

Features like LOTUS 123. But not just a spreadsheet, a program generator as well. Features like Cut and Paste, variable column widths and lightning fast calculation speed. Lucid changes your computer into capacity you never thought possible in the Model 100, 102, $99.95, plus shipping

Disk + ROM

Now your Model 100, 102, 200 plus any desk top computer is an instant disk system. With a function key transfer any files or BASIC or machine code programs to and from your portable and the other computer. The IBM version works over the phone $149.95, plus shipping. Cable $40.00



Business Analyst ROM

Perform those strategic "what if" business problems. Allows detailed financial analyses like Breakeven, IRR, NPV, return on investment, interests costs, PV and others at the touch of a function key. Prints detailed reports. So easy! Also it performs simple everyday calculator functions. Check it out! $99.95, plus shipping. Model 100, 102.

HARDWARE

6-ROM Bank

Access Super ROM, Disk +, Business Analyst and others. The 6-ROM Bank with its 30 hour rechargeable battery pack lets you personally create your ultimate Model 100, 102 system. $199.00, plus shipping.

OTHERS

Custom ROMs

Have your own BASIC program put on to a ROM, or combine your program with a PCSG program. $500.00 minimum order — quantity pricing as low as $15.00 per ROM. Model 100, 102, or 200.

Cassette Programs

PCSG also has cassette programs available at closeout prices. Choose from Tutor+ (typing tutor — $24.98), Type+ (makes your M-100 a memory typewriter — $34.98), Tenky+ (financial calculator — $29.98), Data+ (file manager — $29.98), Sort2+ (sorting routine — $14.98), Write+ (word processor — NEC only — $34.98).

Shipping: $7.50 (U.S.) two days; $15.00 (U.S.) next day

PORTABLE COMPUTER SUPPORT GROUP

4540 Beltway Drive, Dallas, TX 75244 • 1(214) 404-4000 • 1-800-544-4699

Circle 78 on Reader Service card.

Device Timer: A Versatile Timing Program

Make your Tandy computer into one of the world's most expensive timers!

by W. R. Henry

DEVICE.BA is a programmable timer that uses the cassette port of the Model 100. If you need a device that can turn on and off for any purpose, one that can do these maneuvers for specified, timed intervals at specified start and stop times, *DEVICE.BA* will do the job.

Any electrical or electronic circuit controlled by this program MUST be completely isolated from the computer circuitry through use of an optical isolator such as is diagrammed in Figure 2 accompanying this article. Figure 1 shows a complete parts list for such an isolator.

PROGRAM OPERATION

Here is how *DEVICE.BA* works: The program prompts you for start and stop times for the device it controls. When you enter these values, the LCD displays three status boxes. The first box shows the starting time and its status (on/off). The second displays the stop time and its status (on/off). The third box displays real time and shows program activation.

After activation, a signal sounds each minute during the timed interval. At the conclusion of the timed interval, all status boxes change to show that the timed interval is complete. Thus, during the timed interval, the user always knows the exact status of the device being controlled.

The programming is straightforward, and needs no

PC=Radio Shack Cat.# 276-116
Q1=Motorola HEP 59100
R1=22K 1/4 Watt Resistor
IC#1=SN7592

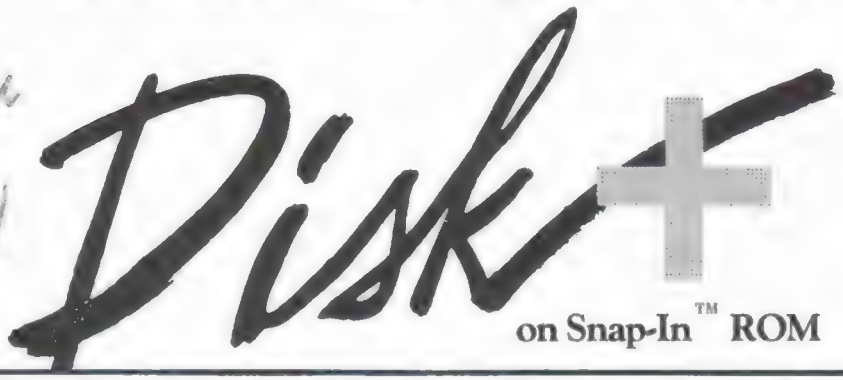
Figure 1. Parts list.

```
Ø 'DEVICE  CASSETTE REMOTE CONTROLLER
1 CLS:CLEAR 1ØØ:PRINT@172,"Motor Control
ler":PRINT@213,"By W.R.Henry":GOSUB4ØØØ:
CLS:'By W.R.Henry*1ØØ3 E.Highway St.*Fre
dericksburg,TX 78624*
2 T$="'''DEVICE TIMER'''":T1$="Start at"
:T2$="Stop at":T3$="Real Time":A$="."'De
fine titles
3 PRINT:INPUT"START TIME (HH:MM:SS)?";S$
'get starting time
4 PRINT:INPUT"STOP TIME  (HH:MM:SS)?";D$
:CLS'get ending time & clr screen
5 LINE (5,12)-(65,5Ø),1,B:'set up title
boxes(line #5 thru 7
6 LINE(9Ø,12)-(15Ø,5Ø),1,B
7 LINE(17Ø,12)-(23Ø,5Ø),1,B
1Ø PRINT@2Ø-LEN(T$)/2,T$'Print title cen
tered
2Ø PRINT@82,T1$:PRINT@97,T2$:PRINT@1Ø9,T
3$'Put titles in boxes
3Ø PRINT@122,S$:PRINT@136,D$'print start
 & end times in boxes
31 PRINT@162,"Status":PRINT@176,"Status"
:PRINT@189,"Status":PRINT@229,CHR$(27);"
pSEQUENCED"'status titles in boxes in re
verse video
35 PRINT@149,TIME$'print real time
4Ø IFTIME$=S$THEN1ØØØELSEIFTIME$=D$THEN
2ØØØELSE35'If start timejump to start ti
me routine,if end time jump to end time
routine else show real time
1ØØØ MOTOR ON:BEEP:BEEP:PRINT@2Ø3,CHR$(2
```

Listing for Cassette Remote Control program.

USE YOUR DESKTOP COMPUTER AS A DISK DRIVE FOR YOUR M-100.

IBM Version works over the phone!



Try *Disk+* for 30 days. If you aren't as excited as we are, return it for a full refund.

When we designed *Disk+* we did it out of necessity. We wanted a way that we could just connect a Model 100 to our desktop computer with a cable and save files onto the desktop's disk drive. We wanted it to be so simple to use it would be self-explanatory.

Picture this. *Disk+* comes to you on a Snap-in ROM and a diskette for your desktop. You take a quarter and open the little compartment on the back of your Model 100. Then you just press the ROM into the socket. *Disk+* appears on your main menu just like a built-in.

You connect your Model 100 to your other computer using an RS232 cable (available from PCSG for $40).

You just place the *Disk+* diskette into the desktop's drive and turn on the computer. It powers up automatically and says "awaiting command" on your desktop's screen. Then you just put the widebar cursor on the Model 100 main menu on *Disk+* and press ENTER. You are shown your RAM files arranged just like the main menu.

To save a file to your other system's disk drive, you just move the widebar cursor to the file you want to save and press ENTER. It is saved instantly with no further action.

To look at the disk directory, you just press a function key on your Model 100. You see immediately the disk directory on your Model 100 screen, and it is arranged just like your Model 100's main menu.

To load a file from the diskette to your Model 100, you just move the widebar cursor to the file and press ENTER. The file is transferred to your Model 100's RAM instantly. You can press F8 and go back to the main menu, and the file you loaded from diskette is there, ready to use.

It is so nice to be able to keep your documents, programs (both BASIC and machine code) and *Lucid* spreadsheet files on the diskette, and bring them back when you need them. All files are ready to run or use with no changes or protocol by you.

If you have access to a desktop computer and don't have *Disk+*, then evidently we have done a poor job telling you about it.

All files and programs that you load or save, go over and come back exactly as they are supposed to be because of full error checking. This guaranteed integrity is really a comfort. *Disk+* is wonderful in so many other ways. For example, you can do a "save all" of all your RAM files with just a touch of a function key. That group of files is saved on the diskette under a single filename with a .SD (for subdirectory) extension. Any time you want, you can bring back all those files at once, or just one or two if you like, again with one-button ease.

Disk+ takes up no RAM. That's zero bytes either for storing the program or for operating overhead.

What really excites most *Disk+* users is text file cross compatibility. Your Model 100's text files are usable on your desktop computer, and your desktop's text files become Model 100 text files.

This means you can write something on your Model 100, and with *Disk+* transfer it instantly to your desktop and start using it right away on your bigger computer. Or the way we like to work is to type in a document on the desktop computer and then transfer it to our Model 100 with *Disk+*. Then we print out the document, beautifully formatted, using WRITE ROM.

Disk+ works with just about every micro sold, from IBM PC and its clones, to all Radio Shack computers (yes, all), to Apple II, Kaypro, Epson and most CPM. Just ask us. More than likely, your computer is supported.

Incidentally, hundreds of Model 100 owners have gone to their Radio Shack stores and bought a color computer because it is so low priced, and with *Disk+* they have an inexpensive disk drive.

And if that weren't enough, how about this: *Disk+* also provides cross-compatibility between different computers like IBM, Apple or the Model 4 using the Model 100 as the intermediary device. Quite a feature!

The snap-in ROM is really great because you can use other ROMs like *Lucid* or WRITE ROM. They snap in and out as easily as an Atari game cartridge and you never lose your files in RAM.

Anyone who ever uses *Disk+* simply can't do without it. But so many times we have had new users call us and say, "Wow! I had no idea when I ordered it that *Disk+* would be so fantastic. I just couldn't believe that I could use my desktop computer's disk drive with my Model 100 just like it is another main menu."

That's why we sell *Disk+* on a thirty-day trial. If you aren't completely satisfied, return it within thirty days for a full refund. Priced at $149.95 on Snap-in ROM. MasterCard, Visa or COD.

(214) 404-4000

PORTABLE COMPUTER SUPPORT GROUP

4540 Beltway Drive, Dallas, TX 75244 • 1(214) 404-4000 • 1-800-544-4699

PCSG provides hotline software support for the Model 100. Call us at 1-(214) 404 4030

Circle 71 on reader service card

documentation for anyone familiar with *BASIC*. I have successfully used this program for timing diverse operations ranging from making my morning coffee to timing test periods for student tests. Readers who need to control external devices should find it useful. Uses are limited only by your needs and your imagination. ❑

```
7)"p>>ON<<":PRINT@229,"ACTIVATED":GOTO35
2000 MOTOR OFF:BEEP:BEEP:BEEP
2010 PRINT@203,">>OFF<<":PRINT@217,">>ON
<<"
2030 PRINT@229,"ENDED AT*":PRINT@157,"*"
:PRINTCHR$(27);"q"
2040 FORP=1TO10:BEEP:NEXTP
2050 FORT=1 TO 2000:NEXT T
2060 BEEP:BEEP:PRINT@280,"RE-SCHEDULE DE
VICE (Y/N)?";
2065 GOSUB 4050:IFA$="Y"ORA$="y"THEN1 E
LSE2080
2080 CLS:PRINT"Thanks for using DEVICE.B
A":END
4000 FORT=1TO1000:NEXTT:RETURN
4050 A$=INKEY$:IFA$=""THEN4050ELSERETU
RN
```

End of listing.

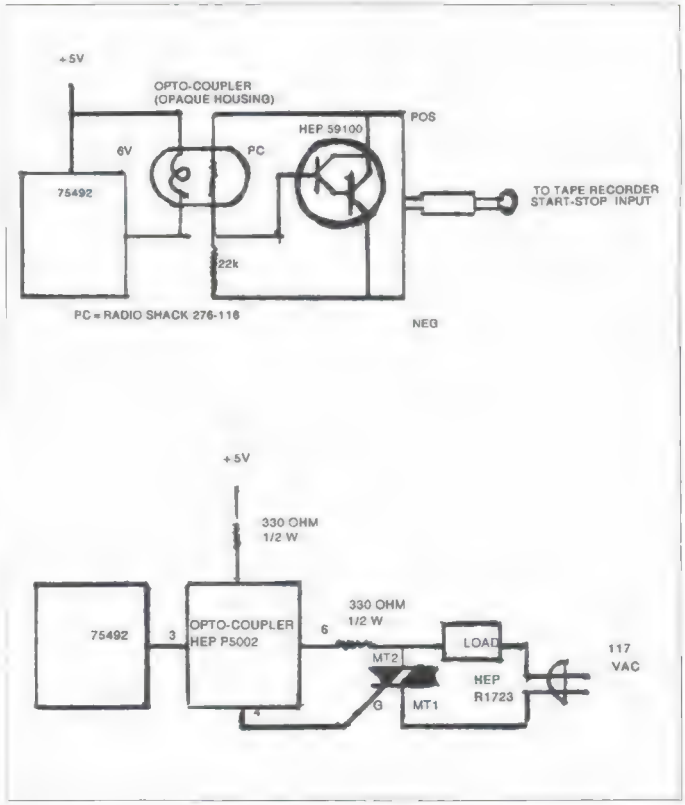


Figure 2. Circuit diagram for remote controlling with your Tandy 100/102.

Two Megabytes & A Dictionary from Traveling Software - *A Review of the Booster Pak*

Is this the ultimate Tandy 100/102 peripheral?

by Stephen R. Lankton

As the owner of a model 102 and two 100's with three RAM expansion modules, a Chipmunk drive, a Tandy drive, a ROM bank, every ROM, a prairie power battery, and God knows what else, I am here to say that you will probably want to replace them all for the Booster Pak. This is the most significant piece of hardware for the 100/102 to date. The reason for this seemingly reckless claim is that the Pak is an answer to many wish lists.

This product is so packed with features that it is hard to decide what to review first. Focusing on the 100, I will begin in the middle.

WHAT IT IS

The Booster Pak for the Tandy 100/102 is a snap-on tray, which includes up to two million bytes of continuous RAM, 64K of ROM software, including Xmodem software, a 1200-baud modem, a ten-hour rechargeable battery pack, a RAM-ROM expansion unit, and the new four-ROM Sardine dictionary, which contains 35,000 words.

Additionally, Traveling Software (TS) has introduced three new methods of controlling so much storage area. Rather than using function keys or software programs to switch between banks of 32K memory storage, as with previous expansions, TS created an innovative way to leave us a single continuous area of two megabytes. All this memory could be difficult to navigate through, however. So TS came up with three clever ideas to avoid the "void": sub-directories, environments and macros.

First, here's a physical description of the Pak.

I'm still showing it to anyone who will look.

.The Booster can be purchased in a basic configuration (136K RAM of which 96K is usable). This adds about 17 ounces of weight and about 3/4 inch to the computer. The Pak is exactly the length and width of the computer and can be held on with four corner-screws or just snapped on with the permanent, 1-1/2 inch wide plastic clips provided. I have merely snapped mine on at this point, because I'm still at the stage of showing it to anyone who will look. The snap seems sufficiently secure.

On the 102, the Pak's added size makes it a quarter of an inch larger than the 100 without modifications. On the 100, the extra depth (2-3/4" in back, 2-1/2" in front) took some getting used to at first. But once my fingers adjusted to the new thickness, I had no complaints, which may have come from the flood of reinforcement my 100 gives me when I began using the Pak.

A fully configured Pak can include all the above features nestled snugly within it. However, cautious buyers might want to begin with the basic unit and work up as their budgets and needs demand. I have the price list at the end of this review for your interest.

USING BOOSTER PAK

Ever wish you would no longer lose programs and data stored on a RAM bank because of a cold boot?

The first lesson TS asks us to learn is the difference between the memory in the Pak itself and the memory available in the Radio Shack unit. The Booster Pak memory, or RAM, is called the RAM disk. Well, it is both RAM and the external storage device—like a disk—therefore RAM disk is a suitable name. Don't confuse this RAM with the RAM in your desktop that can be allocated to an artificial drive. The latter is usually called a RAM disk, also. I guess this is TS's way of distinguishing their RAM from a RAM *bank*.

The Radio Shack internal bank of RAM is called the "workspace," where you do all your computing (with one possible exception I will mention later). To use all this memory, first you must copy files and programs you will be using from the Booster Pak (the RAM disk) to the workspace. When you have done this, one copy of each remains in the Booster Pak until you reverse the procedure and copy files from the workspace back to the Pak. This means that you will never ever have a cold boot that wipes out your data again, a claim that cannot be made by the other makers of RAM modules. This feature alone is just about enough to justify buying it as far as I am concerned.

If you want to look into your ADRS.DO file, for example, all you do is put the cursor on the file name and press *ENTER*. This action copies the file to the workspace, and when you press *F8* to exit the text function, the updated file is copied back to the RAM disk automatically. For setting up a copy of several files to the workspace without opening them as you usually would, you can use *F1* and specify the location of the copy. A file could go to the workspace (by *F2*). It could be sent to the RAM with another name or another location on the RAM disk (by *F1*). Or it could be sent to the Tandy, or compatible, disk drive connected to the serial port (by *F3*). By the way, a copy of the TS disk operating software is provided on the Pak's operating system ROM.

The Booster Pak snapped onto the bottom of the Tandy 102.

SUB-DIRECTORIES

Okay, so here we have a convenient way to get the files in and out of the RAM and the workspace. Ever wish you could switch to groups of programs by a mere push of the cursor? Another wish has come true. Here is a new way to organize all those files: sub-directories. Menus are arranged in a hierarchical way with an entire menu of, say twenty files, at level two represented by a single name on the menu at level one. In the Booster Pak, there are only two levels of directories: the main menu and one sub-level below it.

Those of you familiar with MS-DOS or with the Chipmunk know about sub-directories. For those who are not yet familiar with them, imagine the screen on my 100 when it is turned on. The familiar main menu has the time and date moved to a new location and the *SCHD* and *ADRS* names removed. *BASIC*, *TEXT*, and *TELCOM* still remain and a strange new file name *NULL.##* appears. It also shows several entries which have the extension of .<>, such as *WRITE.<>*, *CALC.<>*, *APTS.<>*, *ARCHIV.<>*, *LETTER.<>*, and so on. These are the names of my sub-directories. I have my files arranged in seven sub-directories. A sub-directory on the Pak can be selected by placing the cursor bar over the name of the sub-directory and pressing *ENTER*—the menu for that sub-directory is then be displayed. You can create an sub-directory instantly by hitting function key *F8* (MKDR) and then typing the desired name. *F8* will also allow you to go from any sub-directory to another without going to the parent or root directory first.

My *WRITE.<>* sub-directory, where this review is stored, contains fifteen additional entries. It too resembles the main menu.

ENVIRONMENTS

Ever wish you could put one program and all the needed files on the computer, get them all off, and put on a new set—and keep it simple? That is a wish answered with Environments. The Booster Pak has the ability to establish, in the laptop, a state that resembles the one you get after a cold boot (turning on the computer while holding *SHIFT-CONTROL-BREAK*). That makes it possible for a program to run without any of the residual interferences from a previous program. This pristine state is called the NULL environment.

At the very least, then, a user could load the NULL environment after using any machine language program or ROM and ensure that the "hooks" left in the RAM registers from the previous program will not cause a problem. This means that something done by Lucid or Write ROM will not cause you to get an out-of-memory error, or worse, force you to reboot if you run TMPC calendar program right after them.

But with the Booster Pak, you can create an environment—all the data files and conditions needed to operate any program properly—and load it automatically with one key stroke.

So along with the usual .DO files and the expected .BA and .CO programs, I have some .## files. I have *SPELL.##*, *WRITE.##*, and *CHECK.##* environments.

Creating an environment is easier than it sounds. First load the NULL environment. Place your cursor on the *NULL.##* file name and press *ENTER*. The computer will ask if you want the NULL environment. An-

swer yes by pressing *Y*. Thus your laptop is in a state that resembles being cold booted—and no data will have been lost. Next, press *F7* to MaKe ENvironment (MKEN). You may enter up to six characters for an environment name, such as *WRITE*. At this point, the environment will remember all the setup you do and duplicate it the next time you load the WRITE environment.

The first thing I am concerned with in my WRITE environment is getting my Ultimate ROM 2 (or for you it might be Super ROM or WRITE ROM) to be a part of the setup. Initially, the Booster Pak responds by presenting a map of the ROM I have installed. I place the cursor over the one I will be using and press *ENTER*. Then I go into *BASIC* and call 63013,1 (as per the instructions with the ROM). Please note, while this sounds dreadfully involved, you do this only one time when you first create the environment and define its contents. Finally, copy all the files you wish to have associated with the WRITE environment into the workspace. That is it.

When you wish to switch environments, or to go to the NULL environment to define still another one, the Pak will ask you if you want to update and essentially save the current configuration of programs, files, memory allocation, etc. If you enter *Y*, the state you left WRITE in will be the state you find waiting for you when you load it again. Since loading the environment thereafter requires only placing the cursor on *WRITE.##* and hitting *ENTER*, the environment becomes a remarkable tool for handling all your customarily used files.

MACRO KEYS

Macros are the next best thing to not working. I am writing this report with View80 in 60 character mode. I will spell-check it with Sardine's 35,000 word dictionary. The method used to pull this file into the Ultimate ROM 2 and set it up was with two keys—*CODE + ENTER*! That is, I turn on the computer, put the cursor over the file and press *CODE* and *ENTER* at the same time. This set of keystrokes is a macro, an automated set of commands that can be triggered with only two strokes. The Booster Pak allows users to set up four different macros of up to fourteen keystrokes for each environment. Since you can have a nearly unlimited number of environments you can have as many macros as you desire. Actually, the limit is determined by the amount of RAM you have installed.

Once you have established the directories and environments you need, setting up the macro is extremely easy. However, the actual decision of what to do with the macro takes some relearning of old habits in some cases. For instance, you do not need to initiate the Ultimate ROM 2 by placing the cursor over it and hitting *ENTER* if you use a macro. I had some difficulty getting the above macro to work because I was doing it wrong. Since I have used the UR2 for many months, I assumed that it worked the same on the Booster Pak when, in fact, it works more efficiently now.

The macro I built begins from the RAM disk menu. When the cursor is over a file I wish to read in View80 and I initiate the *CODE-ENTER* combination, the macro automatically issues a call to the ROM. The UR2 is placed automatically in the workspace and the file is copied to the workspace, too (after hitting *ENTER* with the cursor on the file). So, in one operation, three operations, which I did not have to enter, were automatically done. I taught my macro to go to the View80 option, hit *ENTER*, exit, go to T-WORD, hit *ENTER*, move over to the second file on the menu (which will be the chosen file in every case unless I have copied another file into the workspace first), and hit *F4* to go to the TEXT mode T-WORD. All this is done very quickly and the result is that my file is staring at me in the neat, compressed type of View80. And I never have to type all those strokes again.

When I exit UR2, the file is copied back to the RAM disk space where it is safe from the gremlins that cause cold boots.

USING SARDINE 4-ROM DICTIONARY

A similar routine loads the Sardine four-ROM dictionary from the same sub-directory and gets me ready to do the spell check by hitting *GRAPH + F*. The worst thing about writing on the 100/102 is that the screen size, because it is small, inhibits proper editing. View80 helps solve this problem, but more often than not, I end up with a zillion typos. That is when a good spell checker is needed. TS previously released the Sardine ROM with about 4K worth on it. That was a help. It was not that the dictionary helped spell unknown words, but rather it stopped in the file at the point where a word was not found in its dictionary. So it caught the typos. It caught many correctly spelled words too, having a small dictionary—but that was better than letting typos go by unnoticed.

The files and programs for Sardine four-ROM dictionary interactively work in the special Booster Pak ROM bank. You install the four ROM's into any socket in the Booster Pak. Define an environment (there must be one for each ROM) and make the call from *BASIC*. Then to read a file, load it to the workspace where the Sardine ROM loader will be residing. Macro keys to spell-check files are not possible because the Sardine program begins by entering TEXT, which will clear the keyboard buffer and, apparently, wipe out the macro definitions. (This is also true of some other programs, like Lucid.)

After the file is selected with the WORD+, and EDIT (*F4*) chooses the file, a *GRAPH-F* will begin the dictionary processing. The four-ROM program reads a 15K file in about 1.5 minutes (2235 words). When the file has been read, the word count is clearly displayed and the program also stores the words not found in its dictionary in a temporary file and then displays them for your correcting. You don't have to do anything to initiate this step—it is automatic. At every word not found in the dictionary, Sardine allows you to take action with function keys to accept the current spelling, correct the word by retyping it, view a word in context, look up the word or similarly spelled words in the dictionary, replace the current word with the correct spelling from the dictionary, or save the

word in a user dictionary (WORD.DO) for all future spell checking. When you exit the Sardine program your file is saved back to the RAM disk, and when you exit your SPELL environment, even the WORDS.DO file is saved automatically so the updated file replaces the old one in RAM.

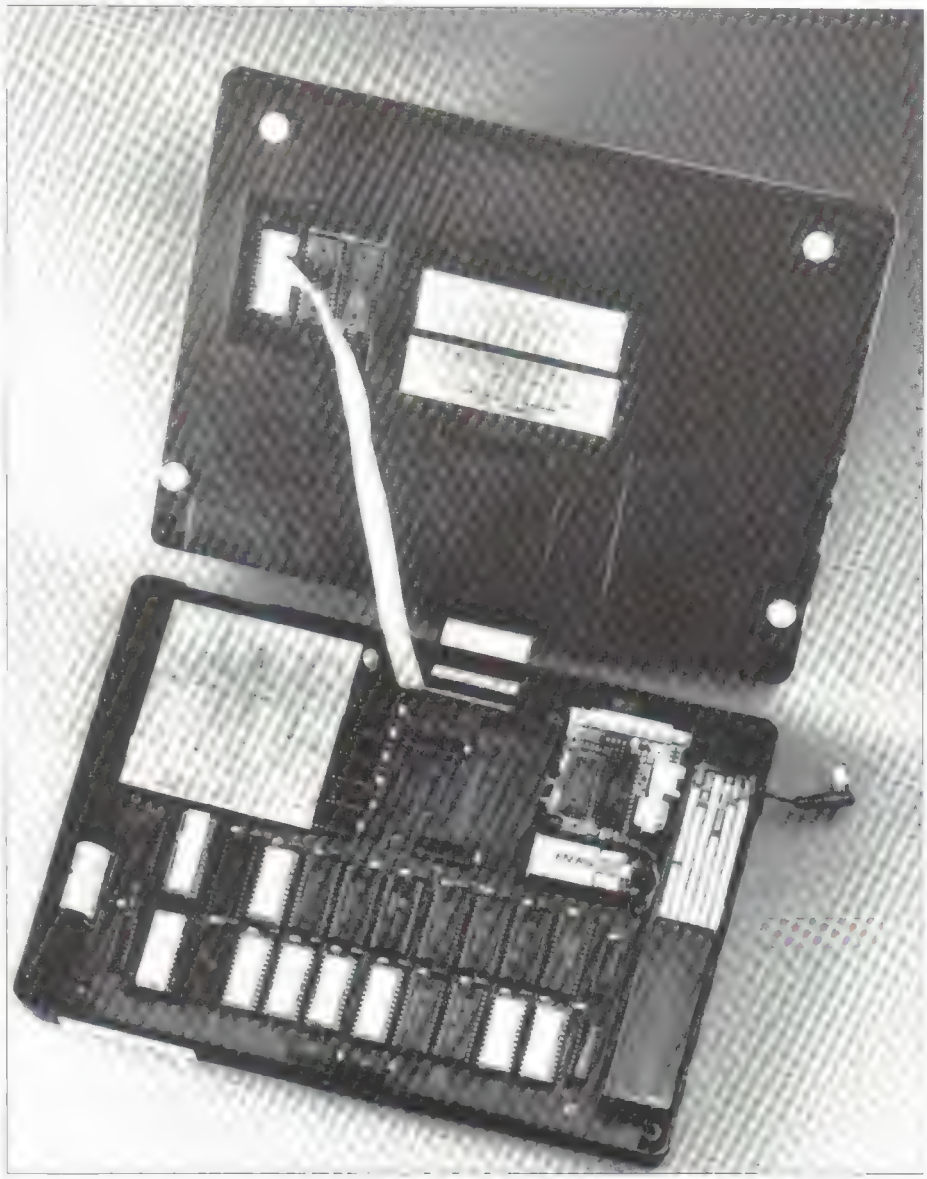
A word on crashes comes to mind here. Let's say you have just about completed the process when the batteries drop out of your 100/102 (this is of course not possible . . . but let's say the Delta flight you are on hits a low pressure pocket, you drop the computer, the Booster Pak falls open, and the batteries roll off under someone else's seat—who knows, it could happen to you). When you get the 100/102 back together, you can bet that the workspace and the file being spell checked would be wiped out. So would any of WORDS.DO and any additions that had been made during that session. But don't despair, these were just copies of the files. The originals would be safe and sound in the Booster Pak RAM. All you would have to do is recopy them or reload the environment and spell check it again. TS techs have convinced me that, although it is technically possible to cold boot the Booster Pak, it isn't going to happen from the operation of a program.

SETTING IT UP INITIALLY

To set up the Booster Pak for the very first time, you do have to cold boot the 100/102 and the RAM Disk. This is done as follows. Save all the programs you have on the 100/102 that you wish to see again. Hold down *CONTROL-SHIFT-PAUSE* and turn the unit off and on once. Go into *BASIC* and type *OUT 5,0* to connect yourself to the ROM port. Then, still in *BASIC*, type *CALL 911* and hit *GRAPH-CONTROL-SHIFT* and *ENTER* at the same time. This will cause the Booster Pak to pause for a moment and initialize the RAM, load the operating system from ROM, and to display the other two ROM programs that come with the Pak: X-TEL and ASTEROIDS.

You may find your computer workspace rebooted for some reason when the Booster Pak is connected, as I mentioned. I have. To re-enter the Pak and continue business as usual, you need only to cold start the 100/102, go to *BASIC* and type *OUT 5,0, ENTER, CALL 911*, and *ENTER*. Notice that you hit *ENTER* to regain entry to the Pak and not *GRAPH-SHIFT-CONTROL-ENTER*.

You will probably never have to do that reboot sequence again. If you install more RAM later by adding the expansion board to go above 256K, you do not have to reboot the Pak. However, if you add further RAM to an already installed RAM board (say to go from 1 MB to 2 MB) you would probably have to save all the files on the RAM disk and reboot it fully.



The Booster Pak plugged into the bottom of the computer.

Saving all the files on RAM disk is an easy matter. If you have a Tandy or Purple Computing disk drive just hit *F5* from the main menu and all the files in the Booster Pak will be backed up automatically. You can also choose to back up only the specific directory of your choice. If you have a desktop in your office, TS makes a Desk-Link communication package that allows you to transfer to or from the disk of the desktop as if it were the Tandy disk drive itself.

X-TEL SOFTWARE AND THE 1200 BAUD-MODEM

But the Booster Pak comes with a terrific communication software

called X-TEL. I have used X-TEL for years now (see "Traveling with Your Portable," *Portable 100/102*, April, '86). It is the only communication package for the 100/102 that allows you to transmit or receive any kind of file or program by phone. That is, any CO, BA, CA, DO DT, etc. file can be sent through X-TEL's Xmodem transfer mode.

It was originally made to interface with the Sigea Systems Telecommuter program for desktops and it works better than you might expect on any BBS. For instance, when you are online with CompuServe, you can go to the TEXT mode and compose a response to EMAIL with all the block move features of the 100/102, go back to the TERM mode, still be on line, and upload the new message. It also is possible to view your directory or delete files while you are online. But perhaps best of all, you can transmit and receive files, which can be any length depending on your memory, directly to the Booster Pak. A simple SPLIT program can break the longer text into a size that you can work with in the workspace, and the COPY command built into the Pak allows you to rejoin the sections of files back into the larger whole for retransmissions.

The Booster Pak comes with X-TEL software and can be purchased with an internal, 1200-baud modem. Let's look at each of these features too, because these make the package really complete. The modem snuggles between the ten-hour battery and the cable connector boards. It is held in with velcro patches and has a nine-volt battery to power it. A cable comes out from the back to the Pak and pulls tightly to allow the modem to plug into the RS-232 connector at the rear of the 100/102. This is a snug fit! The connector comes off if you wish, so you can put on a null modem cable and connect to your desktop. It is a square-cornered standard plug, and it hurts my palm when I try to carry the 100 as a notebook in my usual way. I have to carry it with the front (keyboard) edge in my palm. This is a small price to pay for having communications four times faster than before.

You connect the modem to a modular phone line with a conventional modular phone cable—in other words, you can unplug the four-prong modular connector from the telephone and snap it into the back of the Booster Pak. To use X-TEL with the modem, you enter *X-TEL* from the RAM disk (it is not necessary to copy it to the workspace) and hit TERM (*F4*) to initiate terminal mode. Then, since the modem is Hayes compatible, type *ATDT* (the attention/dial command), the number *1-904-932-6819*, and *ENTER*. For me, next, my desktop will answer with the Telecommuter prompt.

USE OF TANDY DISK

As mentioned above, the TS disk software is provided as part of the Booster Pak system. This allows the movement of all the disk, just subdirectories, or single or tagged files to and from the disk. The tagging of files for deleting or copying is one feature of Booster Pak that further increases its convenience. Tagging is accomplished by pressing *T* when the cursor is on a file. After tagging, any of the commands for file manipulation can be achieved on those files. For example, you could tag several files and copy them to the workspace with a single touch of *F1*. This tag feature is one of several little handy menu-level keystroke aids. You can also hit *P* to print a file under the cursor or *L* to look into the file. Pressing *L* will actually scroll the file past the LCD until you abort it with *ESC*. These are just little extras that keep making the Pak an answer to still other small wishes.

TS has fulfilled my wish list.

FUTURE DEVELOPMENTS

With all that RAM available and the ability to have Xmodem software directly on it we might expect that some later version of T-BASE or T-WORD will access files directly from RAM even if they are larger than the 29K you can squeeze into the workspace. I have heard rumors about that very thing, and I'll just bet within a couple of months we'll see some advertisements for them.

As it is, the TS seems to have thought of everything with the Booster Pak, and they have fulfilled my wish list (at least for now). This product may seem a little pricey at first glance, but then again, it seems to cost in the same general area as other RAM products. In addition, you have room to grow and expand, features not found on other products, as well as the security of never losing your files again becaue of reboots. Newspaper personnel, law enforcers, traveling writers, college students, and the like are going to love this product. And I am betting many current owners of RAM modules and disk drives will also move up to this product and stick with their reliable 100/102 instead of new, weighty IBM compatible laptops. The Booster Pak and companies like TS have significantly helped extend the life and usefulness of our investment.

UNIT REVIEWED

The unit I own is a 2-MB Booster Pak with modem and Sardine four-ROM dictionary. I have Ultimate Rom II and Lucid on additional ROM's. ❑

MANUFACTURER'S SPECIFICATIONS

Traveling Software
North Creek Corporate Center
19310 N. Creek Parkway
Bothell WA 98011
1(800)323-8008.

Basic unit with 136K RAM (96K usable), $429.
Options include 32K RAM chips at $20 each.
Six-slot expansion board at $69.
256K RAM expansion modules, $159.
1200 baud internal modem, $199.
Internal NiCad Battery pack, $69.
Desk-Link, $39.95.
Sardine 4-ROM dictionary, $99.94.

Cross-bank Pasting

We've covered a lot of new Tandy 200 territory together over the last few months, and this month will be no exception. The use of *POKE*ing to CHAIN across banks along with the movement of the transfer buffer, is just scratching the surface of our new abilities. Last month We CHAINed across banks bringing a part of the *PASTE* buffer.

This month I will present the code to bring across the entire *PASTE* buffer while changing banks. This will be a welcomed feature for those who use their Tandy 200 for writing documents. Cutting from one bank and pasting to a document in another bank must be sorely missed, and I am happy to make it available. The key to this feature is the ability to jump to another bank and do specific routines.

Movement of a data byte across RAM banks is accomplished with the *CALL POKE/PEEK (9BB0H/9BB1H)* as discussed in a previous issue of *Portable 100* (see NOTE). The movement of blocks of data is accomplished by the sub-routine *BLKMOV* (Listing 1). *BLKMOV* is set up by preparing the CPU internal registers prior to calling *BLKMOV*.

The HL registers are set to the starting address of the data to be moved, the DE registers are set to the address (in the destination bank) that the data will be moved to, and the BC registers are set to the length of the block of memory to be moved. The only missing parameter is the destination bank. This is calculated early in the routine and stored at *BLKMOV+2*, and is the only parameter passed to the *BLKMOV* routine without the use of the CPU registers.

Now if you are going to jump to another bank to run specific code, you must make sure that the code is in the other bank before jumping there. And that code must contain the way back to the calling bank, and into the calling program. By designing the program to run in the alternate LCD buffer (described last month), we know that the same addresses in all banks are available without fear of corrupting data files or system variables.

This month I will present the code to bring across the entire PASTE buffer while changing banks.

If the exact same routine exists in both banks (at the same locations), then any switching of RAM banks from within the routine would allow the routine to continue to its conclusion. It is important to arrange for the code always to switch back to the original bank before ending.

Just prior to switching banks, it is important to disable interrupts (*DI*) and save the stack pointer (*SP*). When the banks are switched, the SP in the new bank should be made safe for the new bank. Now, while in the new bank, you can call ROM routines, create or delete files, and manage data in general.

After switching back to the original bank, the saved SP must be restored for proper execution of the rest of the program. WHAT A CONCEPT! So with little further ado, let's go on.

PASTE.CO

This program is accessed by placing the widebar cursor over *PASTE.CO* and pressing *ENTER*. You will be asked for *Destination bank* : and any response other than 2 or 3 will be accepted as *1*. You will instantly be at the *MENU* of the selected bank.

The only difference between using this program and using the FUNCTION KEY is that when *PASTE.CO* is used, the PASTE BUFFER is moved into the destination bank, ready to be pasted into any file. The old PASTE BUFFER is overwritten.

There are two exceptions to this. First, if the bank you are leaving has an empty PASTE BUFFER, then the PASTE BUFFER in the destination bank is left intact. Second, if after clearing the PASTE BUFFER in the destination bank there is not enough RAM to accommodate the incoming PASTE BUFFER, a beep will sound and no data movement will be at-

tempted. In either case, you will switch to the destination bank.

Now let's look at the code. The STEPS listed below are separated by ; (blank lines) in the program listing (see Listing 1).

STEP 1 - the *Destination bank:* message is in ROM so we use that text. We print it and get a user response.
STEP 2 - convert that user response into the 0, 4, or 8 needed.
STEP 3 - store the destination bank parameter into the program where needed.
STEP 4 - if we are in the bank selected, just return to the MENU.
STEP 5 - store the source bank and stack pointer for coming back to this bank.
STEP 6 - determine the length of the PASTE BUFFER. If empty then go to STEP 14, else store the length into the code.
STEP 7 - move STEP 8 thru STEP 11 into destination bank using the identical address.
STEP 8 - switch to destination bank to execute Step 9 thru Step 11.

When the program is broken down into its individual routines, it's not all that complicated.

STEP 9 - clear PASTE BUFFER and reset SP.
STEP 10 - expand PASTE BUFFER to length of incoming PASTE BUFFER. Carry flag is set if OM? condition. Get the address in destination bank where data will be written.
STEP 11 - switch back to original bank.
STEP 12 - restore SP and set up the source/destination parameters.
STEP 13 - beep if OM? condition or move the PASTE BUFFER if there was room.
STEP 14 - now jump to the menu of the destination bank.

As you can see, when the program is broken down into its individual routines, it's not all that complicated. There may very well be some other error checking that could have been done, but this works!

NOTE: In earlier issues the routine for *PEEK*ing (*CALL 9BB1H*) should have had the *DI* (disable interrupts) instruction before the call. It was omitted in the listings submitted. The problem occurs if an interrupt is encountered while the banks are switched. The interrupt pushes the PC (program counter) onto the stack but will use the stack location of the original bank. If there is DATA in that location of the switched bank, then that DATA will be destroyed.

-by Paul Globman

```
;************************************
;*       PASTE.CO BUFFER XFER       *
;*          by Paul Globman         *
;*             Copyright (c) 1988   *
;************************************
beep:   equ 4f45h
insert: equ 82a8h
menu:   equ 67a4h
poke:   equ 9bb0h
;
        org 63900
;
        lxi h,6903h     ;msg"dest bank"
        call 11cch      ;print it
        call 12f7h      ;get dest. bank
        rst 4           ;echo
        sui 30h         ;make binary
;-----
        mov b,a         ;convert from
        mvi a,1         ;bin(1,2,3)
ch1:    add a           ;
        dcr b           ;to (0,4,8)
        jnz ch1
        ani 0ch
;-----
        sta bsw+1       ;store dest bnk
        sta blkmov+2    ;into code 3X.
        sta j_bnk+1     ;parameter mgnt
;
        mov b,a         ;If this is the
        in d8h          ;dest bank then
        ani 0ch         ;just go to the
        cmp b           ;Menu (source
        jz menu         ;bank in A)
;
        sta swbck+1     ;store src bnk
        desp 0          ;store SP for a
        xchg            ;safe return.
        shld setsp+1    ;SP saved at
                        ;loc (setsp+1)
;
        lhld 62112      ;find length of
        push h          ;paste buffer
        call 8268h      ;find eof (hl)
        pop b           ;
        hlmbc           ;
        jz j_bnk        ;pste buf empty
        shld bc+1       ;store len bc+1
;
;
        lxi h,bsw       ;move following
        mov d,h         ;executable
        mov e,l         ;code to dest.
        lxi b,setsp-bsw ;bank (from bsw
        di              ;to setsp).
        call blkmov
```

continued.

Listing 1. The assembly language source code.

```
1 '*************************
2 '*  PASTE.CO             *
3 '*    by Paul Globman    *
4 '*  Copyright (c) 1988   *
5 '*************************
10 FOR I = 63900 TO 64040
20 READ X:POKE I,X:SM=SM+X
30 NEXT
40 IF SM = 18273 THEN 60
50 PRINT"error in data":STOP
60 SAVEM"PASTE",63900,64040,63900
1000 DATA 33,3,105,205,204,17,205,247
1010 DATA 18,231,214,48,71,62,1,135
1020 DATA 5,194,171,249,230,12,50,233
1030 DATA 249,50,22,250,50,16,250,71
1040 DATA 219,216,230,12,184,202,164,103
1050 DATA 50,255,249,56,0,235,34,3
1060 DATA 250,42,160,242,229,205,104,130
1070 DATA 193,8,202,15,250,34,246,249
1080 DATA 33,232,249,84,93,1,26,0
1090 DATA 243,205,20,250,62,0,211,216
1100 DATA 49,112,248,205,213,45,205,71
1110 DATA 76,1,0,0,42,160,242,205
1120 DATA 168,130,62,0,211,216,49,0
1130 DATA 0,235,42,160,242,220,69,79
1140 DATA 212,20,250,14,0,195,144,155
1150 DATA 197,6,0,213,86,227,205,176
1160 DATA 155,209,235,193,35,19,11,121
1170 DATA 176,194,20,250,201
1180 REM       END OF DATA
```

Listing 2. The Basic version of the program.

```
;=====================================
bsw:    mvi a,0          ;sw to dest bnk
        OUT D8h          ;and execute!
;
        lxi sp,63600
        call 2dd5h       ;CLR paste buf
        call 4c47h       ;reset SP
;
BC:     lxi b,0          ;len in BC
        lhld 62112       ;target adrs=HL
        call insert      ;cflag = om?
;
swbck:  mvi a,0          ;sw to src bnk
        OUT D8h          ;and fix stack
;=====================================
setsp:  lxi sp,0         ;SP reset
        xchg             ;dest loc in DE
        lhld 62112       ;srce loc in HL
                         ;len in BC
;
        cc beep          ;beep OM?
        cnc blkmov       ;if room, do it
;
j_bnk:  mvi c,0          ;dest bank
        jmp 9b90h        ;bank switch
;=====================================
;The following code will move a block
;of memory in current bank to any
;location in destination bank.
;Destination bank poked to blkmov+2 by
;earlier code.
;BC=len, HL=source adrs, DE=dest adrs
;
blkmov: push b           ;save counter
        mvi b,0          ;dest bank in B
        push d           ;store dest adr
        mov d,m          ;data byte in D
        xthl             ;dest adr in HL
        call poke        ;POKE it!
        pop d            ;restore DE and
        xchg             ;HL registers
        pop b            ;restore BC:len
;
;       inx h            ;increment ptrs
        inx d            ;to do next.
        dcx b            ;decrement len
        mov a,c          ;and test if B
        ora b            ;and C are 0
        jnz blkmov       ;done if BC = 0
        ret
;=====================================
```

The Portable 100 Classifieds

WANTED: backlighting for Model-200 screen. Robert W. McMullen, 739 Morris Way, Sacramento CA 95864-6171

Teachers, stay organized using your gradebook and our Grade Management System: SETUP.BA CLASS.BA GRADE.BA REPORT.BA. Four under 6K. programs written/used by teachers. $49.00. Specify 100/102/200 Disk PDD1/PDD2 Software by STEFFENS, 1731 William Ct., Little Chute, WI 54140

MODEL 100, 102 200 SUPPORT (Five-Year-Old Club)

Ongoing support for Tandy/Radio Shack Model 100, 102 and 200 owners. International support by mail, voice, and bulletin board. Free on-line weekly newsletter 300/1200 baud, 8-bit, 24hrs/7days, 415-939-1246. Voice recording: 425-937-5039.

For INDEX of public domain library, "how to" bulletins, product reviews, write to: CLUB 100, A Model 100 User Group, 984 Hawthorne Drive, Walnut Creek, CA 94596

TANDY 600 MEMORY-$99. We make the 96K memory expansion boards that fit inside your Tandy 600 Portable. You can expand your 600 yourself to 224K with 2 boards. Complete installation instructions, 90 day warranty, and 30 day money back guarantee. Limited time offer, only $99.00 each if you mention this Ad & Portable 100 Magazine. Check Tandy's price, then call us today, while they last. PURPLE COMPUTING—call 1-800-732-5012 or 805-987-4788 in CA. Visa, M/C, Amex.

Individual looking for person to do custom business programming for Model 100. Need lease program for auto leasing and custom finance program. Contact Charles Rogers, 15800 Bull Run Rd #F260, Miami Lakes, Fla 33014—(305)825-1203 or (305)751-1625

SPC Programs for 100/102 Disk or Cassette-Quality Resources—840 McKinley, Plymouth, MI 48170

$500.00 weekly Mailing Circulars in your sparetime. Send large self-addressed Stamped Envelop for Details To: H. E. Boulware, Route 1 Box 9, Vichy, Mo, 65580.

The Portable 100 Classifieds

HOW TO PLACE A CLASSIFIED AD

Categorize your advertisement (Hardware, Software, Services, Wanted, Etc.) and carefully type your message. We are not responsible for correctly interpreting handwritten advertisements. Phone numbers, street numbers, dimensions, and any abbreviations count as one word. Logos, company or product, are not allowed, neither are display advertisements. Business rates are $1.60 per word; non-business rates (individuals advertising) are $1.30 per word. Add up the cost and send the advertisement copy with a check, money order, Visa, or Mastercard number to: Portable 100 Classifieds, 145 Grove St. Ext., P.O. Box 428, Peterborough, NH 03458-0428, c/o Linda Tiernan. Make checks payable to Portable 100 Classifieds. Include your complete return address and phone number (phone number is printed only if it is included in the advertisement itself). Materials due the 1st of the month, two months prior to the magazine cover date (example: materials for the February issue must be received by December 1st.) Advertisements received after the deadline will appear in the next scheduled issue. Payment must accompany order. No refunds for advertisements that miss deadlines, regardless of reason. We reserve the right to change advertisement categories, and to reject, edit, or cancel any advertisement we deem improper. There are no agency discounts for classifieds.
For faster service call 603-924-7949.

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

Ad Category _____ No.of words x (1.60)(1.30) _____ #Issues to run _____

Name _____

Address _____

City _____ State ____ Zip _____

Total Enclosed x (Ad Cost x Issues) _____ Phone Number _____

Visa/Mastercard Number _____ Exp. Date _____

Signature _____

If more space is needed, please use a separate sheet of paper.

ADVERTISER'S INDEX

Free Information

For free information on products advertised in this issue of Portable 100, locate the Reader Service number corresponding to the advertisement that interests you. Circle the number on the Reader Service Card at the center of the magazine (or on the wrapper protecting the magazine if you are a subscriber) and drop it into the mail. The literature you've requested will be forwarded to you without any obligation. Please allow 3--5 weeks for delivery.

ADVERTISER

** Please contact this advertiser directly

Don't forget to say you saw it in Portable 100!

The sky's the limit...
...with Portable 100 magazine.

On the road . . . at home . . . in the office. . . in the air. . . the variety of tasks you can perform with your Tandy portable is expanding constantly.

To keep fully informed and up-to-date about current trends, new products, and new uses for your laptop, you need **Portable 100** magazine.

From sophisticated input/output (I/O) calls and their applications, to simple disk drives, **Portable 100** magazine covers it all.

Portable 100 gives you features, news, columns, and reviews that are thorough and timely. And they are written in a fast-paced, easy-to-read style, by leading experts in the computer field.

In upcoming issues you will discover. . .

- how new peripherals can make you more productive.
- more efficient ways to communicate with your desktop.
- where to buy low-cost public-domain software.
- how your business can be more profitable with your portable at your fingertips.
- where to find the best bulletin boards and information services.
- and much more!

Don't miss even a single valuable monthly issue. Fill out the coupon below, or to charge it to your credit card, call

1-603-924-7949

Send no money now! We will gladly bill you.

☐ **YES!** I want to be more productive with my Tandy portable... and save 42% off the newsstand price. Send me a year's subscription (monthly except for a combined July/August issue) of Portable 100 for us$24.97.

☐ Payment enclosed ☐ Bill me
Make checks payable to **Portable 100**

Name _____

Address _____

City _____ State ___ Zip _____

Canda and Mexico, us$29.97, Foreign surface, us$44.97, Foreign Air Mail add us$50. All foreign, except Canada, US Funds drawn on US bank. Please allow six weeks for delivery. LTP

MONEY BACK GUARANTEE:
If you are not completely satisfied with **Portable 100**, you may cancel your subscription and receive a full refund. Please allow 6-8 weeks for delivery of your first issue.

SUPER ROM

Lucid Spreadsheet Write ROM Database Outliner

NOW YOU CAN REALLY HAVE IT ALL!

All on one ROM. Truly the finest four programs available for the Model 100 — guaranteed. Try it for 30 days. If you aren't blown away by the excellence return it for a full refund.

$199.95

The four best programs for the Model 100 all on one ROM. 32K of power without using any RAM for program storage. This is the PCSG Snap-In ROM that just presses easily into the little ROM socket in the compartment on the back. You access the four right from the main menu like built-ins.

Write ROM — the definitive word processor for the Model 100. Function key formatting or dot commands. Search and replace. Library feature — inserts words, phrases or whole documents into text from just a code. MAP lets you see a picture of your document. In all there are 60 features and functions. No one can claim faster operation. FORM lets you create interactive forms with on-screen prompts that you can answer from the keyboard. Nothing else for the Model 100 compares with the features of Write ROM. Exactly the same as the Write ROM sold as a single program. Infoworld says it "makes the Model 100 a viable writing unit . . . surpassed our highest expectations for quality and clarity."

Lucid Spreadsheet: This is the one PICO magazine says "blows Multiplan right out of the socket" and Infoworld performance rated as "excellent" and said "makes the Model 100 compute." Gives you features you cannot get with Lotus 123. Lets you build spreadsheets in your Model 100 that would consume 140-150K on a desktop. Program generating capability with no programming knowledge required. Variable column widths. Includes find and sort with function key control. It's fast, recalculates like lightning. No feature has been taken from the original, only new ones added.

Database: This is a relational data base like no other. You can do everything from mailing lists to invoices. No complicated pseudo-coding, you create input screens as simply as typing into TEXT. You are not limited by size; you can have as large an input screen as you wish. Prints out reports or forms, getting information from as many files as you like. Complete math between fields. Total interface with Lucid worksheets.

Outliner: Does everything that Thinktank does on a PC but a whole lot better. Includes a Sort for your headlines. Lets you have headlines of up to 240 characters. Has cloning, hoisting and sideways scroll up to 250 characters. Like Lucid, this one sets a new standard for outliners. This is the way to plan and organize your projects.

Present Lucid and Write ROM owners can upgrade for $150. If you have both it's $125.

As usual PCSG sells the Super ROM on a thirty day guarantee. If for any reason you are not satisfied, simply return it for a full refund.

We are excited about this product. Super ROM gives the Model 100 the true power of a desktop. No other multi-program ROM has software that compares. But don't take our word for it. We invite you to make that comparison yourself. Priced at $199.95 on Snap-In ROM.

Got stuck with somebody else's multi-ROM? We'll upgrade it for $150.

(214) 404-4000

PORTABLE COMPUTER SUPPORT GROUP

4540 Beltway Drive, Dallas, TX 75244 • 1(214) 404-4000 • 1-800-544-4699

MC, Visa, American Express, Check, or C.O.D.

Circle 40 on Reader Service card.